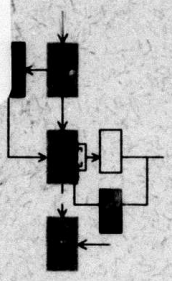
LEVEL II

January, 1980

LIDS-TH-963

Research Supported By:
Grant DARPA/ONR-N00014-75-C-1183
(82933)
Grant NSF/ENG-77-19971
(86035)



A UNIFIED THEORY OF FLOW CONTROL AND ROUTING IN DATA COMMUNICATION NETWORKS

Seyyed Jamaleddin Golestaani

Laboratory for Information and Decision Systems
Formerly
Electronic Systems Laboratory
MASSACHUSETTS INSTITUTE OF TECHNOLOGY, CAMBRIDGE, MASSACHUSETTS 02139

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. 3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) (6) A UNIFIED THEORY OF FLOW CONTROL AND ROUTING IN DATA COMMUNICATION NETWORKS	5. TYPE OF REPORT & PERIOD COVERED Technical
	6. PERFORMING ORG. REPORT NUMBER (14) LIDS-TH-963
7. AUTHOR(s) (10) Seyyed Jamaaloddin Golestaani	8. CONTRACT OR GRANT NUMBER(s) (15) ARPA Order No. 3045/5-7-75 ONR N00014-75-C-1183 w/ARPA Order - 3045
9. PERFORMING ORGANIZATION NAME AND ADDRESS Massachusetts Institute of Technology Laboratory for Information and Decision Systems Cambridge, Massachusetts 02139	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Program Code No. 5T10 ONR Identifying No. 049-383
11. CONTROLLING OFFICE NAME AND ADDRESS (11) Defense Advanced Research Projects Agency 1400 Wilson Boulevard Arlington, Virginia 22209	12. REPORT DATE Jan[illegible] 1980
	13. NUMBER OF PAGES 97
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Office of Naval Research Information Systems Program Code 437 Arlington, Virginia 22217 (12) 100	15. SECURITY CLASS. (of this report) UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

(9) Doctoral thesis,

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Flow Control Networks Data Communication

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A joint flow control and routing (JFCR) strategy is proposed for store and forward communication networks. The strategy is based on a convex optimization problem in terms of the average input rates and multi-commodity flows and is shown to have the following properties: First the average load of each buffer stays below some arbitrarily chosen level for the input rate and routing assignments of the strategy. This level can be chosen so as to upper bound the probability of buffer overflow arbitrarily. Secondly, by proper

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE S/N 0102-LF-014-6601

410 950

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20.
selection of the cost function, it is possible to utilize the network fully and to achieve a variety of different types of priorities in the services offered to the users. Finally, the routing assignments of the strategy correspond to a routing strategy which tends to minimize the total delay when the network is lightly loaded and tends to prevent congestion when it is heavily loaded. Furthermore, the proposed JFCR problem is shown to be equivalent to a minimum delay routing problem corresponding to a bigger network. Accordingly, any minimum delay routing algorithm can be converted into a JFCR algorithm. Using this approach, a class of JFCR algorithms with distributed computations at the nodes are developed.

Under certain conditions, a one to one correspondence is shown to exist in a store and forward network between the set of average input rates and the set of average number of outstanding packets of commodities. This unique correspondence is used to show that in practice the average input rates can be adjusted as desired by restricting the number of outstanding packets on each commodity (window strategy). It is further shown that the upper bounds (windown sizes) corresponding to each set of input rate assignments can be computed distributively in the network.

If a sufficiently fast algorithm with frequent updates is employed, the JFCR strategy can cope with quasi-static variations of the load offered to the network. On the other hand the window strategy is effective in controlling the dynamic fluctuations of the traffic. Thus the analytical features and quasi-static effectiveness of the JFCR strategy is combined with the fast dynamics and practicality of the window strategy.

January 1980 LIDS-TH-963

A UNIFIED THEORY OF FLOW CONTROL AND ROUTING IN DATA COMMUNICATION NETWORKS

by

Seyyed Jamaaloddin Golestaani

This report is based on the unaltered thesis of Seyeed Jamaalodin Golestaani, sumbitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at the Massachusetts Institute of Technology, January 1980. The research was conducted at the M.I.T. Laboratory for Information and Decision Systems, with support provided in part by grant DARPA/ONR-N00014-75-C-1183 and grant NSF/ENG-77-19971.

Laboratory for Information and Decision Systems
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

A UNIFIED THEORY OF FLOW CONTROL AND ROUTING IN DATA COMMUNICATION NETWORKS

by

SEYYED JAMAALODDIN GOLESTAANI

S.B., Tehran University of Technology
(1973)

S.M., Massachusetts Institute of Technology
(1976)

E.E., Massachusetts Institute of Technology
(1976)

SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Dec. 1979

Signature of the Author S. J. Golestaani
Department of Electrical Engineering and
Computer Science, December 31, 1979

Certified by
Robert G.Gallager
Thesis Supervisor

Accepted by
Arthur C. Smith
Chairman, Departmental Graduate Commitee

A UNIFIED THEORY OF FLOW CONTROL AND ROUTING IN DATA COMMUNICATION NETWORKS

by

SEYYED JAMAALODDIN GOLESTAANI

Submitted to the Department of Electrical Engineering and Computer Science
on Dec. 31, 1979 in partial fulfillment of the
requirements for the Degree of Doctor of Philosophy

ABSTRACT

A joint flow control and routing (JFCR) strategy is proposed for store and forward communication networks. The strategy is based on a convex optimization problem in terms of the average input rates and multi-commodity flows and is shown to have the following properties: First the average load of each buffer stays below some arbitrarily chosen level for the input rate and routing assignments of the strategy. This level can be chosen so as to upper bound the probability of buffer overflow arbitrarily. Secondly, by proper selection of the cost function, it is possible to utilize the network fully and to achieve a variety of different types of priorities in the services offered to the users. Finally, the routing assignments of the strategy correspond to a routing strategy which tends to minimize the total delay when the network is lightly loaded and tends to prevent congestion when it is heavily loaded. Furthermore, the proposed JFCR problem is shown to be equivalent to a minimum delay routing problem corresponding to a bigger network. Accordingly, any minimum delay routing algorithm can be converted into a JFCR algorithm. Using this approach, a class of JFCR algorithms with distributed computations at the nodes are developed.

Under certain conditions, a one to one correspondence is shown to exist in a store and forward network between the set of average input rates and the set of average number of outstanding packets of commodities. This unique correspondence is used to show that in practice the average input rates can be adjusted as desired by restricting the number of outstanding packets on each commodity (window strategy). It is further shown that the upper bounds (window sizes) corresponding to each set of input rate assignments can be computed distributively in the network.

If a sufficiently fast algorithm with frequent updates is employed, the JFCR strategy can cope with quasi-static variations of the load offered to the network. On the other hand the window strategy is effective in controlling the dynamic fluctuations of the traffic. Thus the analytical features and quasi-static effectiveness of the JFCR strategy is combined with the fast dynamics and practicality of the window strategy.

Thesis Supervisor: Robert G. Gallager

Title: Professor of Electrical Engineering and Computer Science

تقدیم به والدینم

DEDICATED TO MY PARENTS

ACKNOWLEDGMENT

I wish to express my sincere gratitude to Professor Robert G. Gallager for his guidance, encouragement and support during the course of this research. As my thesis supervisor, his insight, observations and suggestions helped establish the overall direction of the work reported here. I am also indebted to him for his great help during the course of writing.

I would like to thank Professor S.K. Mitter, Professor J.J. Massey, Professor P. Humblet and Professor D.P. Bertsekas for many valuable discussions. The last two have been especially helpful in reviewing this report as my thesis readers.

Thanks also go to Ms. F. Frolik for her excellent typing of the manuscript and to Mr. A.J. Giordani for the illustrations.

This research was carried out at the MIT Laboratory for Information and Decision Systems with partial support provided by grant DARPA/ONR-N00014-75-C-1183 and grant NSF/ENG-77-19971.

TABLE OF CONTENTS

CHAPTER I - INTRODUCTION

1.1 Description of the Problem

In a store and forward (S/F) data communication network, each node is equipped with some storage capacity, called a buffer. The messages arriving at each node will be queued in this buffer and wait until they can be transmitted over an appropriate outgoing communication link. As in the case of a single server, the sizes of the queues built up in the buffers depend on the rate of the traffic seeking service by the network. As the rate of arriving traffic increases, these queues also will grow in size and the messages will undergo larger delays when passing through the network. Since the nodal storages are limited in size, if the input rates continue to increase, eventually some of the buffers will become congested by the stored data.

When congestion occurs, i.e. when some of the buffers of the network get congested with the data waiting for available links, the efficiency of the network drops since those links which lead to saturated buffers can no longer send data. Therefore, in this situation, an increase in arrival rate leads to a decrease in the service rate. This is an unstable situation and, unless the inputs to the network are reduced sufficiently, will drive the network more into congestion and more buffers will become saturated. As the number of nodes with saturated buffers increases, a situation known as "deadlock" will occur. A deadlock involves several saturated nodes, each one filled up with the data which should be sent to other saturated nodes. In effect, no transmission of data remains possible between these nodes and they will be locked up to each other unless some of the buffered data is dropped out. A deadlock may even involve all of the nodes of the network.

The above comments are roughly illustrated in Fig. 1.1, where the throughput of the network is sketched versus the total stored data in the

buffers. Although the term "throughput" here is a vague notion, since it could be interpreted either as the sum of the throughput of all commodities or sum of the flow of all the links, stillwe find this diagram useful in demonstrating what happens in the network.

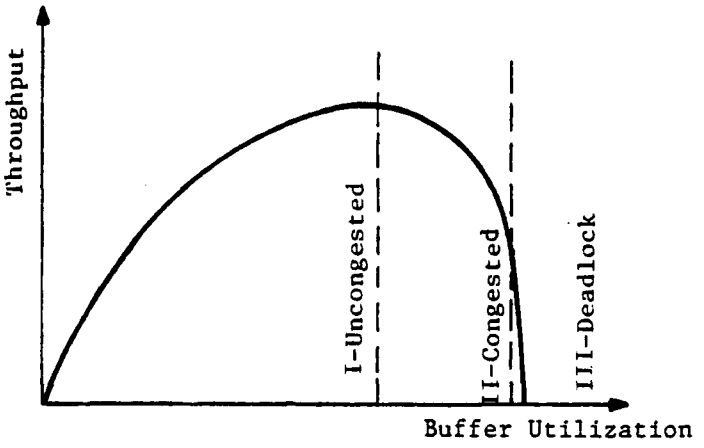


Fig. 1.1

In a well-designed network, the nodal storages and link capacities are properly sized so as to accomodate peak hour traffic requirements and to absorb reasonable short-term fluctuations within the peak hour. This does not guarantee, however, that the input traffic will never exceed the limits of the network. If controls are not imposed, a sufficiently high burst of input traffic can always drive the network into the congestion or deadlock states.

Adaptive routing, flow control and deadlock prevention refer to three types of control schemes which are necessary in order to maintain a small delay and uncongested traffic in the network. A routing strategy, while assuming no control over the rate of arriving traffic,often tries to route the data through the network with the objective of minimizing the total number of messages in the network for the given input rates. When congestion is likely to occur in some parts of the network, adaptive routing can

provide some relief by offering alternate routes to the data passing through the congested regions. However, if the input rates are higher than the maximum flows achievable by the best routing, then congestion will still occur.

Adequate control procedures must, therefore, be developed to regulate input rates and prevent the network from entering the congestion region. The ensemble of such procedures is generally referred to as the flow control strategy. In a well-designed network some additional control means should be available to avoid deadlock if the flow control procedure does not work and the network becomes partially congested. This latter control procedure is referred to as a deadlock prevention mechanism. Here in this report, we are only concerned with the routing and flow control problems with the emphasis on the flow control. A discussion about deadlock prevention can be found in [1], [2].

The problem of routing has been an active research area in the field of data networks in recent years and several static, quasi-static and dynamic routing algorithms using central or distributed computations have been studied [3] - [6]. The objective of most of these algorithms is to minimize the expected number of messages which are in the network. This is equivalent to minimizing the expected delay of messages travelling through the network. This type of routing is called a "minimum delay routing". There is a different type of routing problem, considered by J.R. Yee [6], with the objective of minimizing the congestion of the most congested link, that is to minimize the ratio of flow over capacity (f/C) for the link with the biggest such ratio in the network. This type of routing is called minmax routing. Minmax routing currently appears to be more difficult to implement than minimum delay routing. However, it is probably more meaningful with respect to the congestion problem, since it tries to minimize the congestion over the worst

link. Both types of routing policies, however, frequently lead to very similar routing assignments [7].

While the routing problem is intensively studied, there is little work done in the area of flow control. Our objective in this research initially was to develop a flow control strategy for data communication networks.What we came up with, however, is a unified approach to both routing and flow conrol.

The kind of congestion that we are concerned with in our study here is the saturation of intermediate nodes in a S/F communication network. This is sometimes referred to as store and forward (S/F) congestion. There exists a less fundamental kind of congestion in S/F networks where different packets (sgements) of the same conversation may take alternate routes. These packets must therefore be reassembled in the correct sequence before delivery to the destination. Deadlock may occur if the number of outstanding packets exceed the size of the reassmbly buffer. This is known as reassembly congestion and is less interesting conceptually because it is an isolated problem involving only an interaction between a source and a destination node.For this reason we focus on the more general problem of store and forward congestion and by "flow control" we mean that kind of control necessary to prevent it.

Basically the objectives of a flow control design are as follows:

i) protection against congestion;

ii) minimum reduction of the network throughput and minimum overhead in normal network conditions;

iii) fairness with respect to all network users.

It is important to keep the second objective in mind because otherwise one may shut down all ports of the network in order to protect it against congestion. The third objective becomes important when a nework is conges-

ted and it is necessary to reduce some of the input traffics. The question is which conversation should be reduced first.

1.2 Historical Background and Previous Results

The problem of flow control in S/F networks goes back to early 1970's when the ARPA network was developed to demonstrate the feasibility of S/F networks. Since then there have been many articles written on the subject and several examples of flow control strategies are found in the literature, a few of which are implemented on real networks [8] - [13]. Gerla and Chou give an excellent summary and critique of some of these strategies [14]. Most of the proposed flow control strategies so far have been ad hoc and despite the importance of the subject, it appears that little systematic work has been done to formulate the general problem and to investigate the relationship between network congestion and other important network functions and parameters such as routing strategy, transmission delay, buffer size, etc. However, some of the suggestions have interesting features. Here we refer to some of the work done so far:

Perhaps the most direct forward flow control scheme is the one proposed by D.W. Davies and simulated in National Physical Laboratory (NPL) UK [9]. Here the idea is to keep the total number of outstanding packets in the network below some critical number P. This can be done by circulating P empties (places) through the network. A packet can enter the network only if it can capture one of these empties. The empty will be released once the packet is at the destination. The method does not prevent local congestion, however, since it controls only the total number of packets but not the packet distribution in the network. Proper distribution of the empties through the network in order to maintain a fast and effective service to the packets waiting at the ports is another critical problem.

Another strategy which appears to be the best among the existing ones is the window strategy [14]. Here the number of outstanding packets between each source and destination pair is kept below a given number called the window size. The window size corresponding to each source-destination pair is usually fixed but might be updated based on some flow control table information circulated in the network. The proposal does not describe, however, how the updating procedure could be performed. It is worth noting that even with fixed window sizes, this strategy provides some adaptive flow control since the input rate for each source-destination pair decreases as the number of highly active source-destination pairs increases.Nevertheless this change is not enough to recover the normal performance of the network when it has become congested because of too many active source-destination pairs.

As examples of the few analytical works done in this area, we mention two articles. The first article, by Pennoti and Schwartz [11], considers the effect of the traffic over a set of tandem links on the service offered to other conversations each of which uses only a single link out of this set. Statistical analysis is then used to evaluate the result of applying some window sizes on the internal traffic. But the analysis is limited to a set of tandem links rather than the entire network and in effect establishes only that by reducing the input rate of one commodity, a better service can be offered to the others.

A more interesting flow control analysis is presented by Lam and Reiser [12]. Here flow control is achieved by applying a limit for every node on the percentage of the corresponding buffer which can be engaged by packets entering the network at that node (input buffer limits). Due to the complexities involved, a statistical analysis is performed only for a homo-

geneous case in which there is complete symmetry amongst the nodes with respect to the number of incoming and outgoing links and their capacities, the rate of arriving traffic, the routing parameters and the available buffer. The throughput of the arriving traffic and the probability of buffer overflow is then computed and sketched in terms of the input buffer limit, and a rule of thumb for calculating the best input buffer limit is suggested.

The analysis is successful in demonstrating some of the important trade-offs such as the trade-off between offering service to different users and the trade-off between increasing input traffic and decreasing congestion. It is, however, limited to a completely symmetrical case and also considers a stationary traffic, therefore does not show how the input buffer limits can be updated in accordance with the changing traffic. Furthermore, input buffer limits, while being very simple to implement and relatively easy for statistical analysis, do not seem to provide sufficiently effective means for flow control. This is because when some node j is congested due to the traffic entering at node i, the input at j will be inhibited, but that at the offending node i is unaffected.

1.3 Overview of the Model and Results

A major source of difficulties in almost all of the previous attempts made to formulate a flow control problem is the statistical analysis of the queues of the network, especially when buffers are considered to be limited in size. A fundamental question, therefore, is whether or not we have to introduce the statistical behavior of the network into analysis in order to derive some effective results. Another question with respect to the model is the choice of the flow control variables, namely the parameters which should be controlled in order to maintain an uncongested traffic. Possible

parameters are the number of outstanding packets, the percentage of the buffers engaged by different commodities, or the average input rates. Which one of these, or other, parameters should be considered as flow control variables in order to formulate the problem and/or achieve an effective flow control in practice?

Here, in the present work, we have primarily avoided a statistical analysis of the queues by formulating the problem in terms of the average quantities involved, such as the average flows and the average buffer loads. Also, for the purpose of theoretical development of the problem, we consider the average input rates of different commodities as our flow control variables. After the problem is formulated and solved and basic results with respect to the average behavior of the network is derived, then we are able to consider the statistical fluctuations of the traffic as well and also to propose other means of achieving flow control which are more practical and effective compared to the control of average input rates.

In Chapter II, we formulate the flow control problem together with the routing problem as a convex optimization in terms of the average input rates and the average multi-commodity flows. We shall refer to this formulation as a joint flow control and routing (JFCR) problem and to the resulting flow control and routing policy as a JFCR strategy. The formulation is based on a static model for the network where there is a fixed set of active commodities and the statistics of these commodities are stationary in time. Nevertheless, the strategy is shown to be applicable to a quasi-static situation where both the set of active commodities and the statistics of these commodities change gradually with time.

The fundamental trade-off between offering service to more traffic and avoiding congestion is embodied in the formulation by trading-off

between two sets of cost functions, one set corresponding to the links (one function for each link) and reflecting the level of congestion, the other set corresponding to the commodities (one function for each commodity) and reflecting the magnitude of the restrictions imposed.

The input rate and the routing assignments of the strategy guarantee that the load of each buffer on the average will remain under an arbitrarily chosen level. Given the statistical fluctuations of the traffic and the maximum available buffer at each node, one can choose this level so as to upper bound the probability of buffer overflow arbitrarily. The routing assignments of the strategy correspond to what we may call a minimum congestion routing. The minimum congestion routing lies somewhere between the minimum delay routing and the minmax routing and shares the advantages of both of them: Like the minimum delay routing it is computationally more amenable while having the congestion relief property of the minmax routing.

The cost functions corresponding to the commodities and the links can be arbitrarily chosen from a wide class of convex functions. We show in Chapter II that by choosing appropriate cost functions for the commodities, it is possible to achieve a variety of different types of priorities in the services offered to the users. Specifically, we show that the relative magnitude of the input rate assignments to different commodities and the relative sensitivity of these assignments with respect to the changes in traffic, can be widely modified by changing certain parameters in the cost functions. Furthermore, we show that if the magnitude of the link cost functions are appropriately chosen, the strategy does not go beyond the necessary magnitude in confining the input rates, in order to achieve flow control.

An important feature of the proposed JFCR strategy is that it is equivalent to a minimum delay routing problem corresponding to a bigger net-

work. This equivalence, which is shown in Chapter III, allows us to use any one of the algorithms proposed for the intensively studied routing problem, in order to develop a JFCR algorithm. As an example of such, we have used the distributed routing algorithm of R.G. Gallager [4] to develop a JFCR algorithm using distributed computations at the nodes of the network.

Finally in Chapter IV we show that under certain conditions there always exists a unique correspondence between the set of average input rates of the network on one hand and the set of average number of outstanding packets of commodities on the other hand. This unique correspondence allows us to use the window strategy as the means of achieving the input rate assignments of the JFCR strategy. We further show that the window sizes corresponding to each set of input rate assignments can be computed distributively in the network. The window strategy is effective in controlling the fast fluctuations of the traffic, a desired property that the JFCR stategy lacks. Thus, we are able to combine the nice analytical features and the quasi-static effectiveness of the JFCR strategy with the fast dynamics and practicality of the window strategy.

CHAPTER II

GENERAL FORMULATION OF A JOINT FLOW CONTROL AND ROUTING PROBLEM

Our objective in this chapter is to show how the routing and the flow control problems in a data communication network can be formulated together as a convex optimization problem. In our discussion we consider a store and forward (S/F) packet switching network. However, the idea is quite general and can be used for the design of flow control strategies in other types of data communication networks. After the model of the network is discussed, we shall propose our formulation of the flow control problem and then shall show that the formulation actually complies with our expectations of a sensible flow control scheme.

2.1 The Model

Consider a S/F packet switching network with N nodes. Let N denote the set of nodes in the network:

$$N = \{i | i = 1,...N\}$$

Let a link from node i to node k be represented by (i,k). In order to discuss traffic flow, we distinguish (i,k) from (k,i), but assume that if one exists, the other one does also. Assume that there exists at least one sequence of links connecting any two nodes in the network. Let L be the set of the links of the network:

$$L = \{(i,k) | \text{ a link goes from } i \text{ to } k\}$$

Let L be the total number of the links in the network. We will sometimes specify a link with only one subscript $\ell = 1,...,L$. Let C_{ik} (C_ℓ) denote

the capacity of link (i,k) (link ℓ).

In general there may exist some exchange of data between any pair of nodes i and j in the network. We refer to the stream of data entering the network at node i and destined for node j as commodit (i,j). Clearly commodity (i,j) is different from commodity (j,i). Let C be the set of all potential commodities (or all source-destination pairs) in the network:

$$C = \{(i,j) \mid i,j \in N \ , \ i \neq j\}$$

In practice, the sequence of arrivals of messages of a given commodity (i,j) forms a random process whose statistics may change from time to time. Furthermore, as times goes on some of the active commodities, namely those that have had some nonzero stream of data, may become silent and some of the inactive ones may start transmitting data. Although our major interest is in the JFCR strategy for a quasi-static situation, for the purpose of theoretical development of the problem we consider for the time being a static case where the active commodities are always active and the silent ones always stay inactive. Accondingly we defin C_A as the set of active commodities:

$$C_A = \{(i,j) \mid (i,j) \text{ is an active commodity}\}$$

Furthermore, we assume that the statistics of the active commodities are stationary. That is, they do not change from one time to another. After we have developed a JFCR algorithm with the present model, we shall discuss its application to a quasi-static case in chapter IV. There, both the set of the active commodities and their statistics change slowly with time.

Let us denote by r_{ij} the expected traffic, in bits per second, enter-

ing the network at node i and destined for node j. We shall refer to r_{ij} as the rate of commodity (i,j). The statistical nature of commodity (i,j) can not be completely characterized by a single parameter r_{ij} and there are other parameters of importance such as the variance of the inter-arrival times and the length distribution of the packets. In our model, however, we do not use any other statistical parameter of the commodities explicitly and characterize commodity (i,j) completely by r_{ij}. Nevertheless, the statistical characteristics that we have neglected will influence the behavior of the resulting JFCR strategy implicitly as we shall see.

For the purpose of accomplishing flow control in the network, we assume that it is somehow possible for each node i to set the rate of any active commodity (i,j) to any value r_{ij} which is selected by the flow control strategy in some interval $0 \leq r_{ij} \leq \lambda_{ij}$. The practical mechanism of doing so will be discussed in Chapter IV. Here λ_{ij} denotes the maximum of r_{ij} and the rate at which commodity (i,j) would have entered the network if there was no flow control practiced by node i. We refer to λ_{ij} as the desired rate of commodity (i,j). Since we have considered the stationary case, we assume that λ_{ij} is a fixed value. We take both r_{ij} and λ_{ij} to be zero for $(i,j) \notin C_A$.

Let f_{ik} denote the total expected flow of link (i,k) in bits per second and let $f_{ik}(j)$ denote that part of f_{ik} which is destined for node j. Let s_{ij} denote the total expected traffic, in bits per second, at node i

---

†Assuming that the source of commodity (i,j) is ergodic, the expected traffic for the stationary case considered here can be written as:

$$r_{ij} = \underset{T\to\infty}{\text{l.i.m.}} \frac{1}{T} \cdot (\text{no. of bits of commodity } (i,j) \text{ entering the network in T seconds})$$

r_{ij} can then be measured approximately over a limited time interval T.

destined for node j. Therefore s_{ij} includes both r_{ij} and the traffic from other nodes that is sent through node i for destination j.

We assume that for every link (i,k), there is a buffer space reserved in node i with the capacity of $B_{ik}(max)$ packets. This space is used to store that part of traffic arriving at node i, which should be sent over link (i,k).* If the data arriving at link (i,k) for a limited period of time exceeds the capacity of the link C_{ik}, the buffer starts to fill up. Similarly when the arriving flow is less than C_{ik} bits per second, the buffer starts to get emptied. In the long run, however, since $B_{ik}(max)$ is a limited number, if there is no buffer overflow, the average traffic arriving at link (i,k) is equal to the average traffic leaving it. Similarly despite the short-term fluctuations in the amount of data stored in the buffers, if there is no buffer overflow the expected flow into a node i for a given destination $j \neq i$ is equal to the expected traffic out of the node for that destination, i.e.

$$s_{ij} = r_{ij} + \sum_{m:(m,i)\in L} f_{mi}(j) = \sum_{k:(i,k)\in L} f_{ik}(j) \quad i,j\in N, \quad i \neq j$$

Finally, let t_{ik} demonstrate the expected delay in second per packet on link (i,k) (including queueing delays at the link input) and let D_{ik} denote the expected number of packets waiting at node i for transmission on link (i,k) or being transmitted on link (i,k). According to Little's formula:

$$D_{ik} = \frac{1}{\Gamma} \cdot f_{ik} \cdot t_{ik} \qquad (i,k)\in L$$

where Γ denotes the expected length in bits of a packet in the network. We

* Practically, at each node i, the available buffer can be shared among all the outgoing links (i,k). Therefore, the distinction made between the available buffers of outgoing links at one node is rather artificial and is just for the sake of theoretical developments.

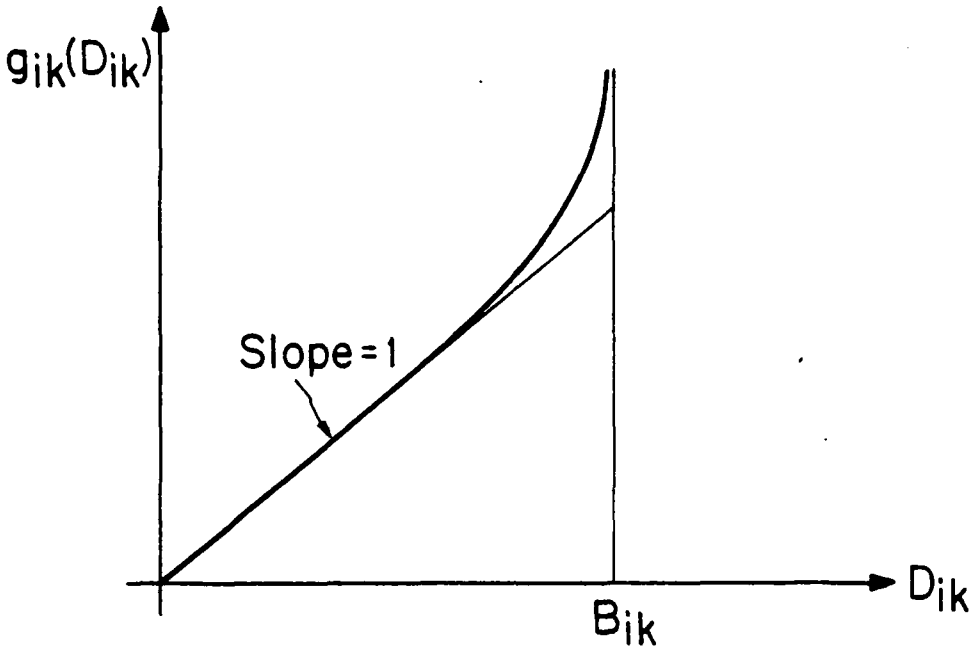
assume that D_{ik} is a function of f_{ik} only. As an example in which this assumption is not a good approximation, consider a network with two commodities which have different packet length distributions. In this case D_{ik} is a function of both f_{ik} and the routing assignments of the network which determine what portion of each commodity will pass through (i,k). However, in order to make the problem analytically tractable, we consider D_{ik} as a function of f_{ik} only.

We shall consider $D_{ik}(f_{ik})$ in its general form, only making the reasonable assumption that it is a convex, increasing and twice differentiable function on the interval $[0,C_{ik})$. Some other notations, which will be used later, are as follows:

- s The set of all node flows s_{ij}; $s = \{s_{ij} | (i,j)\epsilon C\}$
- f The set of all commodity flows $f_{ik}(j)$; $f=\{f_{ik}(j) | (i,k)\epsilon L, (i,j)\epsilon C\}$
- r The set of rates of all active commodities; $r = \{r_{ij} | (i,j)\epsilon C_A\}$
- λ The set of desired rates of all active commodities; $\lambda=\{\lambda_{ij} | (i,j)\epsilon C_A\}$

2.2 Formulation of the Problem as a Convex Optimization

Before we can present our formulation of the problem, we need to discuss two functional quantities which are the core of our strategy. The first quantity, $g_{ik}(f_{ik})$, is a cost function assigned to each one of the links of the network. As long as the average number of packets stored at link (i,k), namely $D_{ik}(f_{ik})$, is far below a given critical value B_{ik}, $g_{ik}(f_{ik})$ is equal to $D_{ik}(f_{ik})$. When $D_{ik}(f_{ik})$ gets close to B_{ik}, the cost approaches infinity (Fig. 2.1). B_{ik} is a fixed parameter chosen according to the size of the maximum available buffer at the link, $B_{ik}(max)$; and should be some fraction of it, i.e. $0 < B_{ik} / B_{ik}(max) < 1$. Therefore, the cost of a link is the



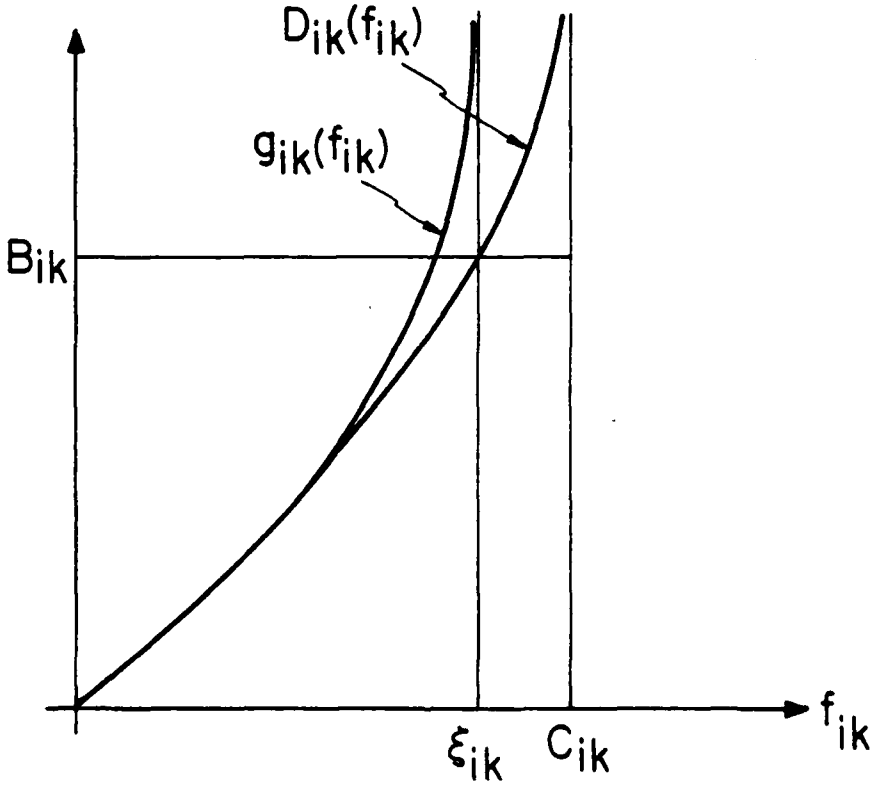


Fig. 2.1

same as its average delay as long as the link is far from becoming saturated. As the load gets heavier and the average quantity of stored data approaches the limit point B_{ik}, the cost builds up rapidly.

Definition 2.1 To each one of the links of the network, there is a cost function $g_{ik}(D_{ik})$ assigned with the following properties:

i) $g_{ik}(D_{ik}) \simeq D_{ik}$ $\qquad D_{ik} << B_{ik}$

ii) $\lim_{D_{ik} \to B_{ik}} g_{ik}(D_{ik}) = \infty$

iii) $g_{ik}(D_{ik})$ is a convex and increasing function on $[0, B_{ik})$,

iv) $g_{ik}(D_{ik})$ is twice differentiable on $[0, B_{ik})$.

We define $g_{ik}(D_{ik})$ to be infinity also for $D_{ik} > B_{ik}$. Properties (iii) and (iv) are added for the sake of subsequent mathematical developments.

Since g_{ik} is a function of D_{ik}, which is a function of f_{ik}, g_{ik} is indirectly a function only of f_{ik}. With some abuse of notation we refer to this new function as $g_{ik}(f_{ik})$. The following lemma is an immediate result of definition 2.1 and the assumptions made about $D_{ik}(f_{ik})$ in Sec.2.1:

Lemma 2.1: For each link (i,k) there is a number ξ_{ik}, $0 < \xi_{ik} < C_{ik}$, for which $\lim_{f_{ik} \to \xi_{ik}} g_{ik}(f_{ik}) = \infty$ and $g_{ik}(f_{ik})$ is convex, increasing and twice differentiable on $[0, \xi_{ik})$ (Fig. 2.1). For $f_{ik} \geq \xi_{ik}$, $g_{ik}(f_{ik})$ is infinite.

The proof is easy and left to the reader.

We refer to ξ_{ik} as the effective capacity of link(i,k) and to $\zeta_{ik} = \xi_{ik}/C_{ik}$ as the efficiency of the link. From now on, we describe a

link as saturated when its average flow has reached its effective capacity and reserve the work "congested" for a link with its total buffer size $B_{ik}(max)$ filled up by the incoming data. Therefore,"congestion"refers to a determinate buffer overflow at some link while "saturation" is a statistical measure about a link.

In the strategy to be discussed later, the effect of the cost function assigned to each link is to prevent the link from becoming saturated. In order to prevent this saturation, the active commodities will have their assigned rates reduced. This in turn requires preventing the assigned rate r_{ij}, $(i,j)\varepsilon C_A$, from becoming too small; thus we should introduce some cost on the amount of restriction imposed on r_{ij} by the flow control strategy. This explains the motive for the following definition:

Definition 2.2: To every active commodity of the network, $(i,j)\varepsilon C_A$, we assign a cost function $e_{ij}(r_{ij})$ with either of the following properties:

a - $e_{ij}(r_{ij})$ is a positive, decreasing, twice differentiable and strictly convex function on $[0,\infty)$(Fig.2.2.a).

b - $e_{ij}(r_{ij})$ is a positive, decreasing, twice differentiable and strictly convex function on $(0,\infty)$. Furthermore $\lim_{r_{ij}\to 0} e_{ij}(r_{ij}) = \infty$ (Fig. 2.2.b).

We will refer to these two types of commodity cost functions as the cost function without or with singularity at point zero. Throughout our discussion in this report we usually consider the commodity cost functions without singularity at point zero. Wherever we consider cost functions with signularity at point zero we shall specify that.

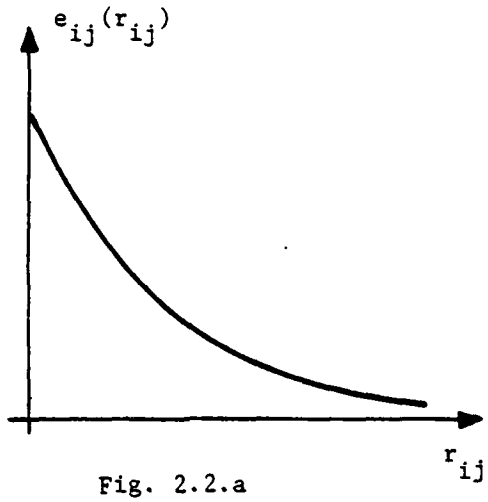


Fig. 2.2.a

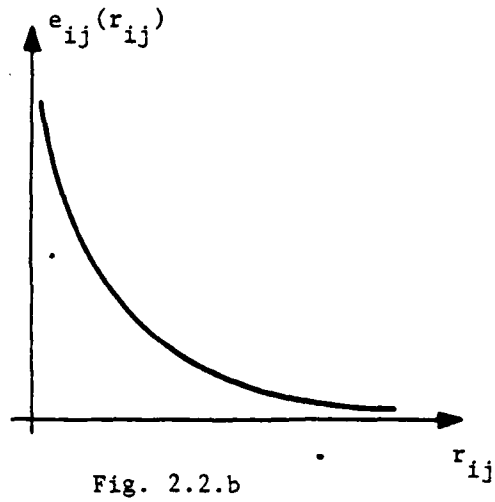


Fig. 2.2.b

Fig. 2.2

At this point we can define the total transmission cost and the total flow control cost of the network, respectively, as follows:

$$G_T = \sum_{(i,k)\in L} g_{ik}(f_{ik})$$

$$E_T = \sum_{(i,j)\in C_A} e_{ij}(r_{ij})$$

Our joint flow control and routing formulation follows immediately:

$$\min_{f,r} \quad J = E_T + G_T \tag{2.1}$$

$$f_{ik}(j) \geq 0 \qquad i \neq j,\ (i,k)\in L \tag{2.1.a}$$

$$r_{ij} \geq 0 \qquad (i,j)\in C_A \tag{2.1.b}$$

$$r_{ij} \leq \lambda_{ij} \qquad (i,j)\in C_A \tag{2.1.c}$$

$$\sum_{\substack{j=1 \\ j\neq i}}^{N} f_{ik}(j) \leq \xi_{ik} \qquad (i,k)\in L \tag{2.1.d}$$

$$\sum_{k:(i,k)\in L} f_{ik}(j) - \sum_{\ell:(\ell,i)\in L} f_{\ell i}(j) = r_{ij} \qquad (i,j)\in C \tag{2.1.e}$$

To verify that the solution to this problem presents a set of desired inputs r and link flows f for the purpose of flow control and routing in the network, we proceed in the following steps:

i) J is a convex function of the variables (r,f) on the set of feasible points defined by constraints (2.1.a) - 2.1.e) (which is a convex set).

ii) The objective function J becomes infinite if $f_{ik} = \xi_{ik}$, i.e. $D_{ik} = B_{ik}$, for some link (i,k). J has a finite value if the rates of all of the active commodities are sufficiently small so that all of the link flows stay below the corresponding effective capacities.

iii) (i) and (ii) imply that (2.1) has at least one optimal point (r^*,f^*) and at any ptimal point the objective function J is limited and the constraint (2.1.d) is inactive.

iv) The value of r in any optimal solution to (2.1) is unique. We show this by contrapositive proof. Suppose that there are two optimal points (r^1,f^1) and (r^2,f^2), with the corresponding objective function J^*. Since E_T is a strictly convex function of r and G_T is a convex function of f, for any $0 < \lambda < 1$ we have:

$$J[\lambda r^1 + (1 - \lambda)r^2,\ \lambda f^1 + (1 - \lambda)f^2] = E_T[\lambda r^1 + (1 - \lambda)r^2] +$$

$$G_T[\lambda f^1 + (1 - \lambda)f^2] < \lambda[E_T(r^1) + G_T(f^1)] + (1 - \lambda)[E_T(r^2) + G_T(f^2)]$$

$$= \lambda \cdot J(r^1,f^1) + (1 - \lambda)J(r^2,f^2) = J^*$$

This contradicts the assumption that (r^1,f^1) and (r^2,f^2) are optimal points. Therefore r^* at any optimal point (r^*,f^*) is unique.

v) Furthermore, (ii) implies that at the optimal point none of the links is saturated, i.e. $D_{ik} < B_{ik}$ for all $(i,k) \in L$. This means that the average number of packets waiting on each link (i,k) can be kept below any

desired number B_{ik}. Nevertheless, due to the statistical variations in the buffer, there will still be some chance of buffer overflow (congestion) at the link. By making the ratio $B_{ik}/B_{ik}(max)$ small enough, one can reduce the probability of buffer overflow arbitrarily. But there are costs incurred in reducing $B_{ik}/B_{ik}(max)$, since this means either an increase in $B_{ik}(max)$ which is costly or a decrease in B_{ik} which reduces the effective capacity of the link.

In summary, the JFCR strategy effectively controls the average load of the buffers, which is to some extent helpful in preventing congestion in the network. In order to prevent congestion more effectively, we need to control the statistical fluctuations of the buffer loads as well. Therefore other means of control should be implemented together with the present scheme. We will investigate this issue in more details in chapter IV.

vi) At an optimal point (r^*,f^*), f^* defines a set of minimum cost routing flows for the set of inputs r^*, with the cost assignements $g_{ik}(f_{ik})$ for the links. This cost is almost equal to the delay of the link, $D_{ik}(f_{ik})$, as long as f_{ik} is far below the effective capacity ξ_{ik}; as f_{ik} approaches the effective capacity, $g_{ik} \to \infty$ (Fig. 2.1). This means that the data will first be routed with the objective of minimizing the total delay. But as soon as some of the buffers get close to the saturation level, the routing will be adapted with the objective of sending data over the nonsaturated links. Despite this apparent difference between minimum delay routing and the routing in our strategy, there exists a basic similarity between them. The following example presents an interesting case where this similarity is most evident.

Example 2.1 Assume a network with M/M/1 queues at all of the links and let

us choose the cost function $g_{ik}(D_{ik}) = \frac{B_{ik} \cdot D_{ik}}{B_{ik} - D_{ik}}$ for all $(i,k)\varepsilon L$. Therefore we have

$$D_{ik}(f_{ik}) = \frac{f_{ik}}{C_{ik} - f_{ik}}$$

and $$g_{ik}(f_{ik}) = \zeta_{ik} \frac{f_{ik}}{\xi_{ik} - f_{ik}}, \text{ where } \zeta_{ik} = \frac{\xi_{ik}}{C_{ik}} = \frac{B_{ik}}{1 + B_{ik}}$$

Let B_{ik} be a constant B for all of the links. Then

$$g_{ik}(f_{ik}) = \zeta \frac{f_{ik}}{\xi_{ik} - f_{ik}}, \quad \zeta = \frac{B}{1+B} = \text{constant}.$$

Since ζ is equal for all of the links, it is easy to see that the above cost function will lead to the same routing if they are changed to $\hat{g}_{ik}(f_{ik}) = \frac{1}{\zeta} \cdot g_{ik}(f_{ik}) = \frac{f_{ik}}{\xi_{ik} - f_{ik}}$. This means that in the given network, according to our strategy, the set of optimal inputs r^* will be routed as if the objective was minimum delay where the capacities are reduced by a factor of $\zeta = \frac{B}{B+1}$ ||.

We should add that if the cost function of any commodity (i,j) has singularity at point zero, the above conclusions (i - vi) are still valid. This singularity will only imply that the inequality constraint (2.1.c) is always inactive for commodity (i,j) and specifically the optimal value of r_{ij} is nonzero, namely $r^*_{ij} > 0$.

We have seen that all of the optimal points of (2.1) share a unique set of inputs r^*, that the set of inputs to the network is routed in a minimal cost way (quite similar to the minimum delay routing), and that the average buffer levels are all constrained below some desired set of limits B_{ik}. To

complete our argument, all we need is to justify the appropriateness of the set of assigned inputs r^*. The rest of this chapter is devoted to this purpose.

2.3 Necessary and Sufficient Conditions for Optimality

Before obtaining the optimality conditions we need to state the following definitions:

Definition 2.3 Any sequence of links connecting an arbitrary pair of nodes i and j is a route R(i,j). That is

$$R(i,j) = \{(n_{k-1}, n_k),\ k = 1,\ldots,\ell \text{ s.t. } n_o = i,\ n_\ell = j \text{ and } (n_{k-1}, n_k) \in L\}$$

Definition 2.4 A route R(i,j) is called an active route if there is some nonzero traffic, travelling from node i to node j over this route.

Definition 2.5 At a given feasible point (r,f), the length of any route R is the sum of the marginal costs of all of the links contained in R That is

$$\text{length of } R = \sum_{k=1}^{\ell} \frac{dg_{n_{k-1}, n_k}}{df_{n_{k-1}, n_k}}$$

Definition 2.6 At a given feasible point (r,f), the distance of a node i from another node j, λ_{ij}, is the minimum length over all possibel routes connecting i to j.

Notice that λ_{ij} is not necessarily equal to λ_{ji}. We take $\lambda_{jj} = 0$ for $j \in N$.

Definition 2.7 The priority function of any active commodity p_{ij} (r_{ij}), $(i,j) \in C_A$, is the negative of its marginal cost function, i.e.

$$p_{ij}(r_{ij}) = - \frac{de_{ij}(r_{ij})}{d\, r_{ij}} \qquad (i,j)\epsilon C_A$$

It follows from definition 2.2 that $p_{ij}(r_{ij})$ is a positive, strictly decreasing and differentiable function on either $(0,\infty)$ or $[0,\infty)$ depending on whether or not $e_{ij}(r_{ij})$ has a singularity at point zero.

Lemma 2.2 Given an optimal point (r^*,f^*), the length of any active route connecting an arbitrary node i to another node j, is λ_{ij}. Furthermore for $(i,j)\epsilon C_A$

$$\begin{aligned} \lambda_{ij} &= p_{ij}(r^*_{ij}) && r_{ij} > r^*_{ij} > 0 \\ &\geq p_{ij}(r^*_{ij}) && r^*_{ij} = 0 \\ &\leq p_{ij}(r^*_{ij}) && r^*_{ij} = r_{ij} \end{aligned} \qquad (2.2)$$

This lemma is proved in Appendix A. The following corollary is a direct result of definition 2.6 and lemma 2.2:

Corollary 2.1 At any optimal point (r^*,f^*), for any link $(i,k)\epsilon L$ and any node $j\epsilon N$:

$$\begin{aligned} g'_{ik}(f^*_{ik}) + \lambda_{kj} &= \lambda_{ij} && f^*_{ik}(j) > 0 \\ &\geq \lambda_{ij} && f^*_{ik}(j) = 0 \end{aligned} \qquad (2.3)$$

where $g'_{ik}(f_{ik}) = \dfrac{dg_{ik}(f_{ik})}{d\, f_{ik}}$.

Theorem 2.1: The necessary and sufficient conditions for a feasible point (r^*,f^*) to be a minimizing point for (2.1) is that there exists a set of positive number β_{ij} , $i,j \epsilon N$, ($\beta_{jj} = 0$, $j\epsilon N$) such that:

$$\begin{aligned} g'_{ik}(f^*_{ik}) + \beta_{kj} &= \beta_{ij} && f^*_{ik}(j) > 0 \\ &\geq \beta_{ij} && f^*_{ik}(j) = 0 \quad (i,k)\epsilon L,\ j\epsilon N \end{aligned} \qquad (2.4)$$

$$
\begin{aligned}
p_{ij}(r^*_{ij}) &= \beta_{ij} && 0 < r^*_{ij} < \eta_{ij} \\
&\geq \beta_{ij} && r^*_{ij} = \eta_{ij} \\
&\leq \beta_{ij} && r^*_{ij} = 0 \qquad (i,j)\epsilon C_A
\end{aligned} \tag{2.5}
$$

Proof: The necessity follows directly from corollary 2.1 taking $\beta_{ij} = \lambda_{ij}$, $i,j\epsilon N$. The sufficiency is proved in appendix A.

Now consider two nodes i and j for which $s^*_{ij} > 0$. By writing Eq. (2.4) for all of the links of some active route $R(i,j)$ and summing them up, one can see that β_{ij} is equal to λ_{ij}. Similarly if $s^*_{ij} = 0$ for some nodes i and j, we can consider Eq. (2.4) for each of the links of a minimum length route $R(i,j)$ and see that $\beta_{ij} \leq \lambda_{ij}$. These results are stated in the following corollary.

Corollary 2.2 For any set of positive numbers β_{ij}, $i,j\ \epsilon N$, ($\beta_{jj} = 0$, $j\epsilon N$) that satisfies (2.3), we have:

$$
\begin{aligned}
\beta_{ij} &= \lambda_{ij} && s^*_{ij} > 0 \\
&\leq \lambda_{ij} && s^*_{ij} = 0
\end{aligned} \tag{2.6}
$$

Finally we should point out that the above results are still valid if any cost function $e_{ij}(r_{ij})$ has singularity at point zero. In this case, however, we know that $r^*_{ij} > 0$ and the optimality conditions with respect to r_{ij} reduce to a simpler form.

2.4 Utilization of Network Resources

Having the necessary and sufficient conditions for optimality at hand, let us consider the set of optimal inputs r^* and see how they correspond with our expectations of a flow control scheme. A suitable flow control scheme should comply with the following:

i) It should be fair with respect to different users. The scheme should not relieve congestion by imposing restrictions only on some of the users - arbitrarily chosen - and leaving the rest free. It rather should impose restriction on the users either evenly or preferably according to some pre-established set of priorities.

ii) The restrictions imposed on the users should not go beyond the necessary magnitude. In other words, the scheme should tend to confine input flows only when it becomes unavoidable in order to keep the network unsaturated.

In the present section we investigate the validity of the second property in our scheme and leave the discussion of the first property for section 2.5. Consider the following optimization problem:

$$\min_{f,r} \quad J = \sum_{(i,j)\in C_A} e_{ij}(r_{ij}) \tag{2.7}$$

$$f_{ik}(j) \geq 0 \qquad i \neq j,\ (i,k)\in L \tag{2.7.a}$$

$$r_{ij} \geq 0 \qquad (i,j)\in C_A \tag{2.7.b}$$

$$r_{ij} \leq \lambda_{ij} \qquad (i,j)\in C_A \tag{2.7.c}$$

$$\sum_{\substack{j=1 \\ j\neq i}}^{N} f_{ik}(j) \leq \xi_{ik} \qquad (i,k)\in L \tag{2.7.d}$$

$$\sum_{k:(i,k)\in L} f_{ik}(j) - \sum_{\ell:(\ell,i)\in L} f_{\ell i}(j) = r_{ij} \qquad i \neq j \tag{2.7.e}$$

This problem is identical to (2.1) execept that the link cost functions are eliminated from the objective function. However, constraint (2.1.d) is kept here to guarantee that at any optimal point no buffer may become oversaturated. Since $e_{ij}(r_{ij}), (i,j)\in C_A$, are decreasing functions, it is clear that at any optimal

point,none of the input rates can be increased without violating (2.7.c.) or (2.7.d). Therefore the optimization problem (2.7) does not impose restrictions on the users beyond what is necessary to keep the network unsaturated.

Due to the difficulties involved in finding the optimal point, the flow control scheme formulated by (2.7),currently does not appear to be suitable for implementation. Nevertheless, as the following theorem indicates, it can be approximated by the proposed JFCR strategy if a set of appropriate cost functions are used.

Theorem 2.2 Let $g_{ik}(f_{ik})$, $(i,k)\varepsilon L$, and $e_{ij}(r_{ij})$, $(i,j)\varepsilon C_A$,satisfy the conditions of definition (2.1) and (2.2). Let $\{\varepsilon_m\}_{m=1}^{\infty}$ be a decreasing sequence of positive numbers with the limit point zero. Assume that (r^m,f^m) is a solution to problem (2.1) with the cost function $g_{ik}(f_{ik})$ replaced by $g_{ik}^m(f_{ik}) = \varepsilon_m \cdot g_{ik}(f_{ik})$, $(i,k)\varepsilon L$. Then any limit point of the sequence $\{r^m,f^m\}_{m=1}^{\infty}$ is a solution to (2.7).

This theorem is a specific case of the barrier function theorem which is proved in [15]. It shows that by sufficiently decreasing the magnitude of the link cost functions, one can bring the solution of (2.1) arbitrarily close the the boundaries where no more increase is possible on the rate of any commodity. Notice that if one holds r constant and minimizes $J^m = E_T(r) + \sum_{(i,k)\varepsilon L} \varepsilon_m \cdot g_{ik}(f_{ik})$ over f only, then the minimizing f is identical for all $\varepsilon_m > 0$. In other words, the routing objective does not change when the cost functions of the links are all multiplied by $\varepsilon_m > 0$.

Despite what it may seem from the above discussion, a very small (or zero) value of ε_m is not desirable, since one can argue that in practice there is a cost to using each link (i,k) that raises rapidly as f_{ik} approaches

ξ_{ik}, and that this cost should really appear as a trade-off against increasing input rates. This would in effect lead to the optimization problem (2.1). Furthermore, solving problem (2.1) requires much computation if ε_m is too small.

2.5 The Trade-off Between Priority Functions of Different Users

In this section, we study the effect of the priority functions $p_{ij}(r_{ij})$, $(i,j)\varepsilon C_A$, on the assigned set of input rates r^* and investigate the fairness of this set with respect to different users. In particular, we show that in offering the service to the users, a variety of types of priorities between them can be achieved through the appropriate choice of priority functions for each user . In doing this, we restrict ourselves to the following class of priority functions with singularity at point zero:

$$p_{ij}(r_{ij}) = \left(\frac{\alpha_{ij}}{r_{ij}}\right)^{n_{ij}} \qquad n_{ij} \geq 1 \tag{2.8}$$

We refer to α_{ij} and n_{ij} respectively as the priority factor and priority order of the commodity (i,j). Naturally, choosing the priority functions from a more general class may prove to provide additional features compared to what we can obtain from this class.

In our evaluation of the fairness and priority issue, we temporarily eliminate constraint (2.1.c) from problem (2.1) by letting $\lambda_{ij} = \infty$, $(i,j)\varepsilon C_A$; therefore making sure that in the set of optimal input rates r^* all the restrictions are imposed by the flow control scheme and not by the users internal limitations. Therefore, with these assumptions we always have $0 < r_{ij} < \lambda_{ij}$, $(i,j)\varepsilon C_A$. Accordingly (2.5) reduces to

$$p_{ij}(r^*_{ij}) = \lambda_{ij} \qquad (i,j)\varepsilon C_A \tag{2.9}$$

In the following discussion, we explain in several steps the types of priorities which can be achieved using the class of priority functions (2.8):

i) First consider two commodities a and b using priority functions of the class (2.8) with the same priority order $n_a = n_b = n$ and assume that the distance of source-destination nodes is equal for them, i.e. $\lambda_a = \lambda_b$. Under these conditions, from (2.8) and (2.9) we have $\frac{r_a^*}{r_b^*} = \frac{\alpha_a}{\alpha_b}$. Thus, when several commodities of the same priority group (i.e. with equal priority orders) experience similar network conditions (i.e. travel through equal source-destination distances), the assigned throughput of each one is proportional to its priority factor. Therefore, larger priority factors should be assigned to the bigger users.

ii) Now consider two commodities a and b with the same priority order n and priority factor α. In order to explore the effect of the topological distance of the source and destination of a commodity on its assigned throughput, we assume that all of the links have equal marginal cost at the optimal point. In this case if the number of links in an active route of commodity a is m times the number for b, we would have $\lambda_a = m \cdot \lambda_b$. Therefore,

$$p_a(r_a^*) = p_b(r_b^*) \cdot m \Rightarrow \frac{r_a^*}{r_b^*} = m^{-1/n}$$

For $n = 1$, the throughputs are inversely proportional to the source-destination topological distances. As n increases, the throughputs become less sensitive to distance. As an example, for $n = 4$ and $m = 2$, $\frac{r_a^*}{r_b^*} = 0.84$. We conclude that $n = 1$ in some sense gives each user equal resource, where $n \to \infty$ gives each the same throughput.

iii) Finally consider two commodities a and b with priority orders

n_a and n_b and priority factors α_a and α_b. If the set of active commodities changes, for example if some new commodities become active, the optimal input rates r_a^* and r_b^* may also change. We would like to compare the amount of changes that r_a^* and r_b^* undergo when some change of traffic happens in the network and investigate the impact that n_a and n_b might have. Since this comparison in general is complicated, we consider the two commodities under exactly similar network conditions, that is we let both of them have the same source and destination nodes i and j.† In this case it follows from (2.8) and (2.9) that:

$$r_a^* = \alpha_a \cdot \lambda_{ij}^{-1/n_a} \quad \text{and} \quad r_b^* = \alpha_b \cdot \lambda_{ij}^{-1/n_b}$$

Since the change in the traffic will be reflected in r_a^* and r_b^* through λ_{ij}, let us compute $\frac{dr_a^*}{d\lambda_{ij}}$:

$$\frac{dr_a^*}{d\lambda_{ij}} = \alpha_a \cdot \frac{-1}{n_a} \cdot \lambda_{ij}^{-\frac{1}{n_a} - 1}$$

or
$$\frac{dr_a^*}{r_a^*} = \frac{-1}{n_a} \cdot \frac{d\lambda_{ij}}{\lambda_{ij}} \tag{2.10}$$

Similarly
$$\frac{dr_b^*}{r_b^*} = \frac{-1}{n_b} \cdot \frac{d\lambda_{ij}}{\lambda_{ij}} \tag{2.11}$$

Therefore, from (2.10) and (2.11)

$$\frac{dr_a^*}{r_a^*} = \frac{n_b}{n_a} \cdot \frac{dr_b^*}{r_b^*} \tag{2.12}$$

---

† In our model of the network, we considered the traffic between any source-destination pair as being one commodity just for the sake of simplicity in the use of notations. The results obtained here with this assumption all will stay valid if we consider several commodities between any source-destination pair.

Eq. (2.12) means that when the traffic in the network changes, then the ratio of the percentage of change in the rate of one commodity to that of the other is inversely proportional to their priority orders, given that the two commodities experience similar network conditions (have the same source and destination). This does not mean that if we increase the priority orders of some commodities, the assigned input rates of all of them will necessarily become less sensitive to the changes in the network traffic. This is because the comparison was made between two commodities which exist in the network at the same time and not between two which replace each other.

We conclude from the above comparison that in general if there is a combination of commodities with high and low priority orders in the network, as the number of active users goes up, the high priority order users will be pushed back more slowly at the cost of lower priority users being slowed down more rapidly. This is exactly our expectation of a priority service system. A quantitative analysis of the sensitivity of the assigned input rates with respect to the changes in the set of active users or changes in the set of desired input rates λ, requires further study.

We can only add here that the sensitivity of the assigned input rates r^*, with respect to the changes in the desired input rates of those commodities for which $r^*_{ij} < \lambda_{ij}$, is zero. This implies that if a user of the network is not assigned as much throughput as it desires (namely $r^*_{ij} < \lambda_{ij}$ for some (i,j)), it can not provoke any increase in the assigned throughput r^*_{ij} by exaggerating about its desired rate, namely by increasing λ_{ij}. The desired input rate, reported by each user, only upper bounds the assigned throughput and has no other impact. This is a basic and important characteristic of the proposed JFCR strategy which is common to any type of priority function used.

CHAPTER III

SOLUTION OF THE JFCR CONVEX OPTIMIZATION PROBLEM

Our primary goal in this chapter is to show how the convex optimization problem (2.1) can be solved in an iterative way using distributed computations in the network. In the first section, however, we present a rather general approach to the solution of (2.1), by making a simple analogy between problem (2.1) and a minimum delay routing problem in general. This analogy actually reduces the problem to a minimum delay routing problem and shows how any method of obtaining a minimum delay routing can be generalized to a solution of problem (2.1).

In the following sections first we reformulate the JFCR problem (2.1) in terms of some new routing variables(different from f) which then allow us to design some algorithms using distributed computations at the nodes of the network to find the set of optimal inputs and optimal routing for the network.

3.1 Analogy of the JFCR Problem with the Minimum Delay Routing Problem

Consider a data communication network M as modelled in section 2.1 and let us construct a new network $\overline{M}$ by making the following changes in M:

Keep the nodes of M unchanged but add one new link between any pair of nodes (i,j) for which an active commodity exists in M. We denote this newly added link by (i,j') in order to distinguish it from an old link (i,j) which might exist in M (and also $\overline{M}$). Thus notice that j' does not indicate a node different from j but the link (i,j') is different from the link (i,j).

Therefore if $\overline{N}$, $\overline{L}$ and $\overline{C}$ respectively denote the set of nodes, the set of links and the set of commodities of the new network $\overline{M}$ we have

$$\overline{N} = N, \quad \overline{L} = L \cup C_A, \quad \overline{C} = C$$

Let the capacities and the average delays of the links of $\overline{M}$ be as follows:

$$\overline{C}_{ik} = \xi_{ik} \qquad \overline{D}_{ik}(\overline{f}_{ik}) = g_{ik}(\overline{f}_{ik}) \qquad (i,k)\in L \qquad (3.1.a)$$

$$\overline{C}_{ij}' = r_{ij} \qquad \overline{D}_{ij}'(\overline{f}_{ij}') = e_{ij}(r_{ij} - \overline{f}_{ij}') \quad (i,j)\in C_A \qquad (3.1.b)$$

Assume that the cost functions $e_{ij}(r_{ij})$, $(i,j)\in C_A$, have a singularity at point zero. It follows then that $\overline{D}_{ij}'(\overline{f}_{ij}')$, $(i,j)\in C_A$, is a positive, increasing, strictly convex and twice differentiable function on $[0, r_{ij})$ and $\lim_{\overline{f}_{ij}' \to r_{ij}} \overline{D}_{ij}'(\overline{f}_{ij}') = \infty$ (Fig. 3.1). Therefore, $\overline{D}_{ij}'(\overline{f}_{ij}')$ complies with all of the properties assumed for the total delay of a link.

Finally, assume that no flow control is practiced in $\overline{M}$ and the throughput of commodities of $\overline{M}$ is $\overline{r}_{ij} = r_{ij}$, $(i,j)\in C$.

In summary, to construct network $\overline{M}$ we have assigned the desired input rates r_{ij} to each commodity of M and instead have added a new link with capacity r_{ij} between i and j. Now consider the following minimum delay routing problem for network $\overline{M}$:

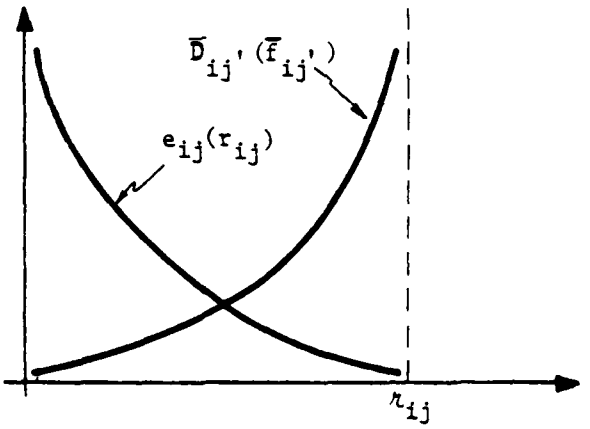


Fig. 3.1

$$\min \bar{D}_T(\bar{f}) = \sum_{(i,k)\in L} \bar{D}_{ik}(\bar{f}_{ik}) + \sum_{(i,j)\in C_A} \bar{D}_{ij'}(\bar{f}_{ij'}) \qquad (3.2)$$

$$\bar{f}_{ik}(j) \geq 0 \quad i \neq j, (i,k)\in L, \; j\in N \qquad (3.2.a)$$

$$\bar{f}_{ij'}(m) \geq 0 \quad i \neq m, \; (i,j)\in C_A, \; m\in N \qquad (3.2.b)$$

$$\sum_{k:(i,k)\in L} \bar{f}_{ik}(j) - \sum_{\ell:(\ell,i)\in L} \bar{f}_{\ell i}(j) = r_{ij} \quad (i,j)\in C \qquad (3.2.c)$$

Notice that the upper limits on link flows are eliminated since given (3.1) they are all inactive at any point with a finite objective value.

Lemma 3.1 : There is at least one optimal point for (3.2) for which each supplementary link (i,j') carries traffic only for destination j, i e.

$$\bar{f}_{ij'}(m) = 0 \qquad (i,j)\in C_A, \; m\in N, \; m \neq j$$

Proof: Let $\bar{f}$ be any optimal point for (3.2). Consider any link (i,j'), $(i,j)\in C_A$, and let $\bar{f}_{ij'}(m) > 0$ for some $m \neq j$. We have $\bar{f}_{ij'}(j) < r_{ij} - \bar{f}_{ij'}(m)$. Therefore, some part of the commodity (i,j), namely some part of $\bar{r}_{ij} = r_{ij}$ is routed to the destination j over some route other than link (i,j'). Call this part $\bar{f}_{ix}(j)$. Obviously $\bar{f}_{ix}(j) = r_{ij} - \bar{f}_{ij'}(j) > \bar{f}_{ij'}(m)$. If we let the flow $\bar{f}_{ij'}(m)$ and an equal part of the flow $\bar{f}_{ix}(j)$ interchange their routes to the node j, the aggregate flows of the network $\bar{M}$, and the total delay $\bar{D}_T$ do not change. Therefore, the new routing is also optimal. Through repeating this interchange of routes for all flows such as $\bar{f}_{ij'}(m)$ we come up with an optimal point satisfying $\bar{f}_{ij'}(m) = 0, (i,j)\in C_A, \; m \neq j$.

Q.E.D.

Corollary 3.1 Any solution to the following routing problem is a solution to (3.2):

$$\min \ \bar{D}_T(\bar{f}) = \sum_{(i,k)\in L} \bar{D}_{ik}(\bar{f}_{ik}) + \sum_{(i,j)\in C_A} \bar{D}_{ij'}(\bar{f}_{ij'}) \qquad (3.3)$$

$$\bar{f}_{ik}(j) \geq 0 \qquad (i,k)\in L,\ j\in N,\ j \neq i \qquad (3.3.a)$$

$$\bar{f}_{ij'}(j) \geq 0 \qquad (i,j)\in C_A \qquad (3.3.b)$$

$$\sum_{k:(i,k)\in L} \bar{f}_{ik}(j) - \sum_{\ell:(\ell,i)\in L} \bar{f}_{\ell i}(j) = r_{ij} \qquad (i,j)\in C \qquad (3.3.c)$$

$$\bar{f}_{ij'}(m) = 0 \qquad (i,j)\in C_A,\ m\in N,\ m \neq j \qquad (3.3.d)$$

Theorem 3.1: Let the cost functions corresponding to the commodities of network M have singularity at zero. The JFCR problem (2.1) for network M then is equivalent to the routing problem (3.3) for network $\bar{M}$. Furthermore, there are solution points to (3.2) which are also optimal for (2.1). The equivalent variables are as follows:

$$\bar{f}_{ik}(j) = f_{ik}(j) \qquad (i,k)\in L,\ j\in N,\ j \neq i$$

$$r_{ij} - \bar{f}_{ij'}(j) = r_{ij} \qquad (i,j)\in C_A$$

Proof: Easy to verify.

The above theorem is important in showing how to solve problem (2.1) by relating it to the routing problem which is intensively studied. It shows that the nature of both problems and their degrees of mathematical complexity are the same, and the JFCR problem only involves a somewhat larger number of variables compared to a routing problem for the same network.

Accordingly, any static or quasi-static approach, using central or distributed computations to the minimum delay routing problem which proves to be successful, can be used for solving the JFCR problem. There are two problems, however, in the blind application of a routing algorithm to solve the JFCR problem. The first is that each dummy link (i,j') should be used only for the (rejected) traffic destined for j. The second and most important is that, both in terms of implementation and convergence, the dummy links should be treated differently than the other links. The next three sections deal with these problems.

3.2 New Formulation of JFRC Problem Aimed at Distributed Solution

Gallager in a recent paper [4] shows how a minimum delay routing problem in the quasi-static case can be solved using only distributed computations at the nodes of the network. The core of his approach is a new set of parameters $\phi_{ik}(j)$, called routing variables, instead of the conventional variables $f_{ik}(j)$, in order to formulate the routing problem. Here, in order to develop a distributed JFCR alogrithm, we shall use the same variables. In the present section, we introduce these new variables and restate JFCR problem (2.1) and the corresponding optimality conditions in terms of them. Then in sections (3.3) and (3.4) we work out some distributed JFCR algorithms. The approaches taken in these two sections are different, however. In section 3.4, we use the analogy made between routing and JFCR problems and, as described earlier, simply use a class of distributed routing algorithms for network $\bar{M}$ to come up with a corresponding class of JFCR algorithms. In section 3.3, however, we take a rather direct approache and, while still using many of the results in Gallager [4], try to keep the distinction between routing and flow control parameters.

Now let us consider the network M as modelled in sec. 2.1. Let $\phi_{ik}(j)$, $(i,k)\in L$, $j \neq i$, denote the fraction of the node flow s_{ij} which is sent over the link (i,k). It follows that

$$f_{ik}(j) = s_{ij} \cdot \phi_{ik}(j) \qquad (i,k)\in L\ ,\ j\in N\ ,\ i \neq j \tag{3.4}$$

$$s_{ij} = r_{ij} + \sum_{k:(k,i)\in L} s_{kj}\, \phi_{ki}(j) \qquad i,j\in N\ ,\ i \neq j \tag{3.5}$$

More formally let us define a routing variable set ϕ for network M as a set of nonnegative numbers $\phi_{ik}(j)$, $i,j,k \in N$, $i \neq j$, satisfying:

a) $\phi_{ik}(j) = 0 \qquad (i,k)\notin L\ ,\ j\in N$

b) $\sum_{k=1}^{N} \phi_{ik}(j) = 1$

c) for each $(i,j)\in C_A$, there is at least one sequence of nodes $i,k,\ell,\ldots,m,j$ such that $\phi_{ik}(j) > 0$, $\phi_{k\ell}(j) > 0, \ldots, \phi_{mj}(j) > 0$.

<u>Theorem 3.2:</u> For any routing variable set ϕ and any set of input rates r, there is a unique set of node flows s and link flows f as the solution to (3.4) and (3.5). Each component s_{ij} or $f_{ik}(j)$ is nonegative and continuously differentiable as a function of r and ϕ.

For proof see Gallager [4]. The above theorem shows that any quantity which can be expressed in terms of (r,f) can also be expressed in terms of (r,ϕ). We can, therefore, restate problem (2.1) as the following:

$$\min_{r,\phi} \ J = E_T(r) + G_T(r,\phi) \tag{3.6}$$

$$r_{ij} \geq 0 \qquad (i,j)\in C_A \tag{3.6.a}$$

$$r_{ij} \leq \hat{r}_{ij} \qquad (i,j)\in C_A \tag{3.6.b}$$

ϕ is a routing variable set (3.6.c)

Constraints 2.1.a. and 2.1.e are inherent in the definition of a routing variable set, as theorem 3.2 establishes; and constraint 2.1.d is inactive at any point with limited objective value. Therefore they are not repeated in problem (3.6).

Theorem 3.3: the following equations are valid and lead to a unique set of solutions for $\frac{\partial G_T}{\partial r_{ij}}$ and $\frac{\partial G_T}{\partial \phi_{ik}(j)}$, which are both continuous in r and ϕ for $(i,k)\varepsilon L$, $j\varepsilon N$, $j \neq i$

$$\frac{\partial G_T}{\partial r_{ij}} = \sum_{k:(i,k)\varepsilon L} \phi_{ik}(j) \left[g'_{ik}(f_{ik}) + \frac{\partial G_T}{\partial r_{kj}} \right] \tag{3.7}$$

taking $\frac{\partial G_T}{\partial r_{jj}} = 0 \qquad j\varepsilon N$

$$\frac{\partial G_T}{\partial \phi_{ik}(j)} = s_{ij} \left[g'_{ik}(f_{ik}) + \frac{\partial G_T}{\partial r_{kj}} \right] \tag{3.8}$$

Proof: Similar results are proved by Gallager [4] for the function $D_T = \sum_{(i,k)\varepsilon L} D_{ik}(f_{ik})$ istead of $G_T = \sum_{(i,k)\varepsilon L} g_{ik}(f_{ik})$. Since $g_{ik}(f_{ik})$ shares all of the properties assumed for $D_{ik}(f_{ik})$, theorem 3.3 is also valid. Q.E.D.

At this point we can state the optimality conditions of the JFCR problem with the new formulation.

Theorem 3.4: The following conditions are sufficient for any feasible point (r^*,ϕ^*) to be a minimizing solution to (3.6):

$$g'_{ik}(f^*_{ik}) + \frac{\partial G_T}{\partial r_{kj}}(r^*,\phi^*) \geq \frac{\partial G_T}{\partial r_{ij}}(f^*,\phi^*) \qquad (i,k)\varepsilon L,\ j\varepsilon N,\ i \neq j \tag{3.9}$$

$$\frac{\partial G_T}{\partial r_{ij}}(r^*,\phi^*) = p_{ij}(r^*_{ij}) \qquad \mu_{ij} > r^*_{ij} > 0$$

$$\geq p_{ij}(r^*_{ij}) \qquad r^*_{ij} = 0$$

$$\leq p_{ij}(r^*_{ij}) \qquad r^*_{ij} = \mu_{ij} \quad (i,j)\epsilon CA \qquad (3.10)$$

where f^* is the set of link flows corresponding to (r^*,ϕ^*). Furthermore, for any optimal point (r^*,f^*) of problem (2.1), there exists some feasible routing variable set ϕ^* such that (3.9) and (3.10) hold true.

Proof: First notice from (3.7) that (3.9) is equivalent to the following:

$$g'_{ik}(f^*_{ik}) + \frac{\partial G_T}{\partial r_{kj}}(r^*,\phi^*) = \frac{\partial G_T}{\partial r_{ij}}(f^*,\phi^*) \qquad \phi^*_{ik}(j) > 0$$

$$\geq \frac{\partial G_T}{\partial r_{ij}}(r^*,\phi^*) \qquad \phi^*_{ik}(j) = 0$$

$$(i,k)\epsilon L,\ j\epsilon N,\ j \neq i \qquad (3.11)$$

Now consider theorem 2.1 and let $\delta_{ij} = \frac{\partial G_T}{\partial r_{ij}}(r^*,\phi^*)$, $i,j\epsilon N$, $i \neq j$. The sufficiency of the above conditions follows directly. To show the second part of the theorem, let (r^*,f^*) be any optimal point for (2.1). Consider any pair of nodes i and j, $i \neq j$, with $s^*_{ij} > 0$. According to lemma 2.2, the length of any active route $R(i,j)$ is λ_{ij}. It follows from (3.7) that $\frac{\partial G_T}{\partial r_{ij}}(r^*,\phi^*) = \lambda_{ij}$ for $s^*_{ij} > 0$. If $s^*_{ij} = 0$ for some i and j, $i \neq j$. $\phi^*_{ik}(j)$ is not determined by f^*. By allowing $\phi^*_{ik}(j)$ to be nonzero only if (i,k) is located on some route $R(i,j)$ with the length λ_{ij}, we will get $\frac{\partial G_T}{\partial r_{ij}}(r^*,f^*) = \lambda_{ij}$ for $s^*_{ij} = 0$ also. Again considering theorem 2.1 and

taking $\beta_{ij} = \lambda_{ij}$, $i,j \epsilon N$, $i \neq j$, it follows that (3,9) and (3.10) are valid valid for (r^*, ϕ^*).

3.3 A Distributed JFCR Algorithm - Direct Approach

Let (r,ϕ) be any feasible point of (3.6). We define the algorithm A as the product of two algorithms A_r and A_ϕ, i.e.

$$A = A_\phi \cdot A_r \tag{3.12}$$

Algorithm A_r only changes r, while algorithm A_ϕ only changes ϕ. The mapping $(r^1,\phi) = A_r(r,\phi)$ is a point to point mapping as follows:

$$\delta_{ij} = \frac{\partial J}{\partial r_{ij}} = \frac{\partial G_T}{\partial r_{ij}} - p_{ij}(r_{ij}) \qquad (i,j)\epsilon C_A \tag{3.13}$$

$$r^1_{ij}(r,\phi) = r_{ij} - \mu\delta_{ij} \qquad 0 \leq r_{ij} - \mu\delta_{ij} \leq \pi_{ij}$$

$$= 0 \qquad r_{ij} - \mu\delta_{ij} \leq 0$$

$$= \pi_{ij} \qquad r_{ij} - \mu\delta_{ij} \geq \pi_{ij} \tag{3.14}$$

where μ is a positive scale factor of A_r to be discussed later.

Before introducing A_ϕ, we have to make the following definition:

Definition 3.1 : Let η be a given positive number. For any routing variable set ϕ and any pair of nodes i, j, $i \neq j$, we define B_{ij} as the set of all nodes $k \epsilon N$ for which either $(i,k) \notin L$ or $\phi_{ik}(j) = 0$ and k is blocked relative to j. A node k is blocked relative to j if there exists a route R(k,j) with the following properties: i) For every link in R(k,j), the routing variable with respect to j is nonzero. ii) R(k,j) contains some link (ℓ,m) for which:

$$\frac{\partial G_T}{\partial r_{mj}} > \frac{\partial G_T}{\partial r_{\ell j}} \tag{3.15}$$

$$\phi_{\ell m}(j) > \eta \left[g'_{\ell m}(f_{\ell m}) + \frac{\partial G_T}{\partial r_{mj}} - \frac{\partial G_T}{\partial r_{\ell j}} \right] \Big/ s_{\ell j} \qquad (3.16)$$

We define another set $\bar{B}_{ij}$ similar to B_{ij} except with ($\geq$) in (3.15) and (3.16).

Now let us define the mapping $(r,\phi^1) = A_\phi(r,\phi)$ as follows:

$$\phi^1_{ik}(j) = 0 \; ; \; \Delta_{ik}(j) = 0 \qquad k \in \hat{B}_{ij} \qquad (3.17)$$

where $\hat{B}_{ij}$ is chosen by node i, in every application of mapping A_ϕ arbitrarily such that $B_{ij} \subseteq \hat{B}_{ij} \subseteq \bar{B}_{ij}$. For $k \notin \hat{B}_{ij}$ define:

$$a_{ik}(j) = g'_{ik}(f_{ik}) + \frac{\partial G_T}{\partial r_{kj}} - \min_{m \notin \hat{B}_{ij}} \left[g'_{im}(f_{im}) + \frac{\partial G_T}{\partial r_{mj}} \right] \qquad (3.18)$$

$$\Delta_{ik}(j) = \min \left[\phi_{ik}(j), \eta \, a_{ik}(j)/s_{ij} \right] \qquad (3.19)$$

where η is a scale parameter of A_ϕ and is the same quantity used in definition 3.1. We shall discuss the proper value of η later. Let $K_{min}(i,j)$ be a set of values of m that achieve the minimization in (3.18). Then:

$$\phi^1_{ik}(j) = \phi_{ik}(j) - \Delta_{ik}(j) \qquad k \notin K_{min}(i,j) \qquad (3.20.a)$$

$$\phi^1_{ik}(j) \geq \phi_{ik}(j) \qquad k \in K_{min}(i,j) \qquad (3.20.b)$$

$$\sum_{j=1}^{N} \phi^1_{ik}(j) = 1 \qquad (3.20.c)$$

Notice that $\phi^1(r,\phi)$ as defined by Eq. (3.17) - (3.20) is not unique since $\hat{B}_{ij}$ is not generally unique and also for a given $\hat{B}_{ij}$, $K_{min}(i,j)$ can have more than one element. Therefore the mapping A_ϕ and $A = A_\phi \cdot A_r$ are point to set mappings.

The algorithm A_ϕ, as defined here, is a modification of the distributed minimum delay routing algorithm proposed by Gallager [4]. There is an error in the proof of lemma 6, appendix C of [4]. The modification of A_ϕ and some respective changes in the proof of convergence is suggested by Gallager as corrections to [4]. The JFCR algorithm $A = A_\phi \cdot A_r$ proposed here is a generalization of the routing algorithm A_ϕ.

Theorem 3.5 Let (r^o, ϕ^o) be any feasible point of (3.6) and let $J(r^o,\phi^o) \leq J_o$. for each value J_o, there exist scale factors μ and η for A_r and A_ϕ such that any sequence $\{(r^m,\phi^m) \varepsilon A(r^{m-1},\phi^{m-1})\}_{m=1}^{\infty}$ converges to a solution of (3.6).

This is proved in appendix B. The next thing is to see whether the computations necessary for this algorithm can be conducted distributively at the nodes of the network instead of being performed at a central node. It turns out that a condition known as loop-freedom is essential for distributed computations of the algorithm to be possible. Therefore, first we shall define the concept of loop-freedom:

Definition 3.2 : A set of routing variables ϕ is called loop-free if for any destination j,there is no directed loop in the network with links all having nonzero routing variables with respect to the destination j.

Theorem 3.6: If (r^o,ϕ^o) is loop-free (namely if ϕ^o is loop-free) so is $(r^o,\phi^1) \varepsilon A_\phi (r^o,\phi^o)$, for any value of the scale factor η in the algorithm A_ϕ.

This theorem is proved in [4]. We just point out here that the introduction of the set $\hat{B}_{ij}$ in the algorithm A_ϕ was necessary in order to establish this result. From lemma 3.2 it follows that if (r^o,ϕ^o) is loop-

free, any $(r^1,\phi^1) \in A(r^o,\phi^o)$ is also loop-free since A_r does not change the routing variables. Therefore, by induction, $(r^m,\phi^m) = A^m(r^o,\phi^o)$ is loop-free for $m = 1,2,\ldots,$ In other words if we start with a loop-free point, then the routing remains loop-free at all of the stages of the algorithm A.

Now let the routing variables be loop-free at the starting point (and therefore at all of the stages) of the algorithm A. In order to demonstrate how the algorithm may be performed using distributed computations at the nodes of the network, let us first show the method of computing $\frac{\partial G_T}{\partial r_{ij}}$, $i,j \in N$, $i \neq j$, distributively. Consider any node i and destination $j \neq i$. Let $\frac{\partial G_T}{\partial r_{mj}}$ be known at all nodes m for which $\phi_{im}(j) > 0$ and assume that all such nodes m send the value of $\frac{\partial G_T}{\partial r_{mj}}$ to node i (over link (m,i). Once node i receives $\frac{\partial G_T}{\partial r_{mj}}$ for all m with $\phi_{im}(j) > 0$, it can compute $\frac{\partial G_T}{\partial r_{ij}}$ from Eq. (3.7). This process of computing $\frac{\partial G_T}{\partial r_{ij}}$ in fact can be started at nodes i which send all the traffic s_{ij} directly to j (namely all nodes i for which $\phi_{ij}(j) = 1$), since $\frac{\partial G_T}{\partial r_{jj}}$ is zero. Then the computation can be done for the nodes i which send the traffic s_{ij} either directly to j or to the nodes of previous class. This process can continue until $\frac{\partial G_T}{\partial r_{ij}}$ is known at all nodes $i \in N$. For every destination j, a separate process is necessary. The property of loop-freedom is necessary to avoid deadlock situations where $\frac{\partial G_T}{\partial r_{ij}}$ should be known at some node i before it can be computed at another node m, and vice-versa, so that both nodes i and m wait indefinitely for the other.

Each node i, in the process of computing $\frac{\partial G_T}{\partial r_{ij}}$, will also get the value of $\frac{\partial G_T}{\partial r_{mj}} + g'_{im}(f_{im})$ for all $m \in B_{ij}$. Therefore, once $\frac{\partial G_T}{\partial r_{ij}}$ is known at all nodes $i \in N$ for every destination $j \in N$, the mapping A_r or A_ϕ can be

applied on the current (r,ϕ) and every node i can compute the new values of r_{ij} or $\phi_{ik}(j)$, $k: (i,k)\in L$, $j \neq i$. Notice that in the algorithm A proposed here, each of the two parts A_r and A_ϕ should be applied on the network in separate iterations. It is certainly desirable to update r and ϕ both at the same iteration. However, the proposed algorithm A has this limitation since the objective here was to show that the JFCR problem can be solved using iterative and distributed computations, and not to find the fastest algorithm.

3.4 A Class of Distributed JFCR Algorithms

Bertsekas [16] and Gafni [17] have recently generalized Gallager's distributed routing algorithm. Their formulations allow the use of second derivative information and also provide the flexibility for our purpose of treating the dummy links of section 3.1 differently from the other links. We have applied the results of section 3.1 to the class of algorithms in [17] to come up with the following class of distributed JFCR algorithms. The equivalence of this class of algorithms with the routing algorithms of [16] and [17] is shown in appendix B.

Consider the network M as modelled in section 2.1 and let us define a set of JFCR variables ψ as following:

$$\psi_{ik}(j) = f_{ik}(j) \Big/ \bar{s}_{ij} \qquad i,j\in N,\ i \neq j,\ (i,k)\in L \qquad (3.21.a)$$

$$\psi_{ij'}(j) = [r_{ij} - r_{ij}] \Big/ \bar{s}_{ij} \qquad (i,j)\in C_A \qquad (3.21.b)$$

where $$\bar{s}_{ij} = r_{ij} + \sum_{m:(m,i)\in L} f_{mi}(j) \qquad (3.21.c)$$

It follows from (3.21) that:

$$\psi_{ik}(j) \geq 0\ ,\ \sum_{k:(i,k)\in L} \psi_{ik}(j) = 1 \qquad i,j\in N,\ i \neq j,\ (i,j)\notin C_A \qquad (3.22.a)$$

$$\psi_{ik}(j) \geq 0, \quad \psi_{ij'}(j) \geq 0, \quad \psi_{ij'}(j) + \sum_{k:\,(i,k)\in L} \psi_{ik}(j) = 1, \quad (i,j)\in C_A \qquad (3.22.b)$$

Definition 3.3 Let ψ be any set of variables $\psi_{ik}(j)$ which satisfy Eq. (3.22). If for any $(i,j)\in C_A$, there is a sequence of nodes $i,k,\ell,..\ m,j$ for which $\psi_{ik}(j) > 0,\ \psi_{k\ell}(j) > 0,..\ \psi_{mj}(j) > 0$, then ψ is called a JFCR variable set.

Theorem 3.7: Given the set of desired input rates λ, any set of JFCR variables ψ corresponds to a unique set of input rates r and multicommodity flows f.

This lemma is proved in appendix B. Now let

$$\gamma_{ik}(j) = g'_{ik}(f_{ik}) + \frac{\partial G_T}{\partial r_{kj}} \qquad i, j\in N, \quad i \neq j, \quad (i,k)\in L$$

For any pair of nodes i and j, $i \neq j$, define two column vectors ψ_{ij} and γ_{ij} respectively with entries $\psi_{ik}(j)$ and $\gamma_{ik}(j)$, $(i,k)\in L$, in any fixed order. If $(i,j)\in C_A$, we extend the dimension of ψ_{ij} and γ_{ij} by one to include respectively $\psi_{ij'}(j)$ and $p_{ij}(r_{ij})$ as their last entry.

For any set of JFCR variables ψ and any node i, we define a node k as blocked with respect to destination j if $(i,k)\in L$ and $\psi_{ik}(j) = 0$ and either $\frac{\partial G_T}{\partial r_{ij}} \leq \frac{\partial G_T}{\partial r_{kj}}$ or there exists a route $R(k,j)$, with nonzero JFCR variables on each part it, which contains some link (ℓ,m) with $\frac{\partial G_T}{\partial r_{\ell j}} \leq \frac{\partial G_T}{\partial r_{mj}}$. For any given ψ, we denote by B_{ij} the set of all blocked nodes with respect to node i and destination j.

Now the clas of JFCR algorithms A_ψ are as follows: In each iteration, in order to map the current point ψ into the new point $A_\psi(\psi)$, at each node i and for every destination j the following optimization problem

is solved

$$\min_{\hat{\psi}} \quad \gamma_{ij}^t (\hat{\psi}_{ij} - \psi_{ij}) + \frac{1}{2} \frac{\bar{s}_{ij}}{\alpha} (\hat{\psi}_{ij} - \psi_{ij})^t . M_{ij} (\hat{\psi}_{ij} - \psi_{ij})$$

subject to i) $\hat{\psi}_{ij}$ satisfies Eq. (3.22)

ii) $\hat{\psi}_{ik}(j) = 0$ for $k \varepsilon \mathcal{B}_{ij}$

where $\alpha > 0$ is a scale factor and M_{ij} is a symmetric matrix of proper dimension that can be a function of ψ and the iteration, but must satisfy the following constraints for some fixed $\Lambda > 0$, $\varepsilon > 0$: First the elements of M_{ij}, say $m_{p\ell}$, must satisfy $|m_{p\ell}| \leq \Lambda$. Secondly M_{ij} should satisfy $\varepsilon ||v||^2 \leq v^t . M_{ij} \cdot v$ for all vectors v of proper dimension which have a zero component on places corresponding to nodes $k \varepsilon \mathcal{B}_{ij}$.

Theorem 3.8: Assume that $\lim_{r_{ij} \to 0} e_{ij}(r_{ij}) = \infty$, $i,j \varepsilon \mathcal{C}_A$. Let ψ^o be any JFCR variable set corresponding to a feasible point (r^o, ϕ^o) with $J(r^o, \phi^o) \leq J_o$. For each value of J_o, there exists a positive value for α such that any sequence $\{\psi^m = A_\psi^m(\psi^o)\}$ converges to a solution of (2.1). Furthermore, if ψ^o corresponds to a loop-free routing, so does ψ^m, $m = 1,2,\ldots$.

This theorem is proved in appendix B. Since the JFCR variable set ψ is loop-free at every stage of the algorithm, the distributed computation of A_ψ is possible.

CHAPTER IV

THE USE OF WINDOW STRATEGY FOR IMPLEMENTATION OF JFCR STRATEGY

In this chapter we discuss three distinct problems regarding the implementation of the JFCR strategy and its effectiveness in controlling the flow of traffic in the network. The first problem is that of adjusting the input rates of different commodities to the values assigned by the JFCR strategy. We prove that a flow control mechanism known as widow strategy - which has been in the literature for some time - is an effective way of adjusting the input rates to the set of assigned values.

Next, we discuss the behavior of the JFCR algorithm in a quasi-static situation where the statistics of arriving traffic changes slowly with time. Finally, we consider the short-term fluctuations of the traffic and show that while the JFCR strategy itself is not capable of reacting to the fast fluctuations of the traffic, the window strategy employed for the input rate adjustment, effectively reduces the danger of congestion caused by fast changes of traffic.

4.1 The Window Strategy for Input Rate Adjustment

The strategy developed in the foregoinf chapters, aims at maintaining an appropriate and noncongesting traffic in the network, through assigning a set of optimal input rates to the commodities. It is simply assumed there that the average rate of the incoming messages on each commodity (i,j) can be set to any value on some interval $[0,r_{ij}]$, and the difficulties involved in the actual implementation are not considered. Our objective in this section is to propose an appropriate mechanism for adjusting the input

rates to the assigned values.

Consider any commodity (i,j) with the desired input rate λ_{ij} and the allowed throughput $r_{ij} \leq \lambda_{ij}$ assigned by the flow control strategy. Accordingly, there must be some controlling device at the entrance of node i which will accept some of the incoming traffic into the network and reject the rest.[+] It is,therefore,necessary to define the rules according to which the decision about acceptance or rejection of the incoming messages should be made in order to maintain the input rate r_{ij}.[++] The window strategy is a flow control scheme which provides such rules. In the present section, first we explain what this strategy is and then discuss its application for our purpose.

Let us first define a term which will be used in the explanation of the window strategy. At a given time an "outstanding packet" in the network is one which has already entered the network and either it has not yet arrived at the destination or its acknowledgment has not yet been received by the source node.

The window strategy refers to a control scheme in which each source node i, keeps the number of the outstanding messages of each commodity (i,j) below a given number w_{ij}, called the window size of commodity (i,j).

[+] If storage capacity is available, the rejected traffic can be queued at the entrance of the network until it can be accepted into the network.

[++] The arrival rate of commodity (i,j) at node i, is not equal to the desired input rate λ_{ij} in general. It depends on the nature of the source and on the value of r_{ij} as well as λ_{ij}. For example, if a human being is at the source of the commodity, a small assigned throughput r_{ij} can discourage him from sending data. On the other hand, if the source is a computer which consistently tries to send some data, the arrival rate would include both the traffic that is arriving for the first time and the traffic that has not gone through before, and therefore is bigger than λ_{ij}.

In any case,it is reasonable for our purpose to assume that the arrival rate is greater than or equal to r_{ij} for $r_{ij} \leq \lambda_{ij}$.

Accordingly, whenever a new message arrives for commodity (i,j), it can enter the network if the number of outstanding packets of commodity (i,j) at that time is less than w_{ij}. If this number is equal to w_{ij}, no packet of commodity (i,j) can enter the network until an acknowledgment from node j is received by i, indicating that a new packet of this commodity has reached the destination.

To our knowledge this strategy was first proposed by Cerf and Kahn [10] and later on discussed by Gerla and Chou [14] as a mechanism for congestion control in data communication networks.It was argued that a set of appropriate window sizes for different commodities would provide an effective flow control in the network. The proposals, however, did not say much about the criteria or computation of an appropriate set of window sizes.

Here we link the window strategy and the JFCR strategy together, by showing that it is possible to maintain any assigned set of input rates r (which is demanded at some stage of the JFCR algorithm) through implementing the window strategy with an appropriate set of window sizes. In other words, we show that for any set of feasible input rates r, there exists a set of window sizes $w = \{w_{ij} | (i,j)\epsilon C_A\}$, such that the implementation of the window strategy using these windows will force the input rates to become equal to r.

We will spend the rest of this section verifying the above claim. In doing so, first we need to obtain the relationship between the average number of outstanding packets and the rate of commodities in the network. Later on, in Section 4.4, we will discuss the interesting features of the window strategy and explain why we have proposed it for maintaining the desired set of input rates in the network.

Consider the network N with some given set of routing variables ϕ. Let K be the number of active commodities of the network. For the sake of notational simplicity, in this section we will mostly use a single subscript $k = 1,2,\ldots K$, in order to refer to an active commodity instead of specifying it with its source and destination pair (i,j). Thus r_k, for example, represents the rate of commodity k. Let n_k (n_{ij}) be the expected number of the outstanding packets of commodity k (commodity (i,j)). This number consists of two parts. The first part, which we shall denote by n_k^1, is the average number of packets of commodity k at any given time, which have entered the network and have not reached the destination. The second part, n_k^2, is the average number of packets of the commodity which have reached the destination but whose acknowledgment has not yet reached the source node.

For the given set of routing variables ϕ, let us define the routing matrix Q, with dimensions K x L, as follows: Each component $q_{k\ell}$ of Q denotes the fraction of commodity k which passes through link ℓ.* The ℓ'th column of Q, $q^{\ell} \in \mathbb{R}^K$ specifies the fractions of different commodities passing through link ℓ. Similarly the k'th row of Q, $q_k \in \mathbb{R}^L$ shows how commodity k is routed through the network. With this definition, it follows from Little's formula that the average number of packets of commodity k, waiting or being transmitted on link ℓ, is $\frac{1}{\Gamma} r_k \cdot q_{k\ell} \cdot t_{k\ell}$ for $k = 1,..K$ and $\ell = 1,..,L$, where Γ is the average length in bits of packets and $t_{k\ell}$

---

* In specifying matrix Q in terms of ϕ it is reasonable to assume that the traffic is routed through the network regardless of the source that each packet is originated from and based only on the destination. With this assumption, each routing variable set ϕ specifies the routing table of every single commodity as well as the overall traffic and, therefore, corresponds to a unique routing matrix Q.

denotes the average delay per packet that commodity k undergoes, waiting or being transmitted on link ℓ.

In order to avoid statistical analysis of the queues of the network, which is very hard when flow control is practiced even under many simplifying assumptions, we assume that on each link the average delay per packet experienced by different commodities is the same.[†] Therefore,

$$t_{k\ell} = t_\ell \qquad k = 1,..,K \ , \quad \ell = 1,...,L$$

With this assumption we can conclude from the above results that:

$$n_k^1 = \frac{1}{\Gamma} r_k \sum_{\ell=1}^{L} q_{k\ell} \cdot t_\ell = \frac{1}{\Gamma} \cdot r_k \cdot q_k \cdot \vec{t} \qquad (4.1)$$

where $\vec{t} \in \mathbb{R}^L$ is a column vector consisting of the components t_ℓ.

Instead of writing Little's formula for every single link, we can consider the whole network as a single server for commodity k and apply Little's formula to it to see that:

$$n_k^1 = \frac{1}{\Gamma} \cdot r_k \cdot \tau_k \qquad (4.2)$$

where τ_k represents the average delay per packet for commodity k when travelling through the network. Now by comparing (4.1) and (4.2) we get:

$$\tau_k = q_k \cdot \vec{t} \qquad (4.3)$$

In order to compute n_k^2, let us denote by θ_k the average time between the moment that a packet of commodity k arrives at the destination and the moment that the corresponding acknowledgment is received by the

---

[†] This assumption is not true in general, despite looking trivial. Supposing that the service times of each packet at different links are exponentially distributed and statistically independent, we have been able to use the results of [18] and show that $t_{k\ell} = t_\ell$ for all $\ell = 1,...,L$ and $k = 1,..,K$, if a window strategy is not practiced and the arrival of packets of each commodity is poisson. But if the window strategy is imposed even on some of the commodities, the assumption is not valid. Nevertheless, we think it is a reasonable approximation, especially since an exact analysis has been impossible.

source node. By applying the same concept one can see that:

$$n_k^2 = \frac{1}{\Gamma} r_k \cdot \theta_k \tag{4.4}$$

Therefore: $$n_k = n_k^1 + n_k^2 = \frac{1}{\Gamma} r_k (\tau_k + \vartheta_k) \tag{4.5}$$

In the following two cases we continue our analysis under two different assumptions, with respect to the time necessary for the acknowledgments to reach the source nodes. In both cases it is assumed that the acknowledgments do not add to the traffic of the network. This is reasonable if data packets contain a field for acknowledgment of other packets.

CASE 1 - θ_k is fixed and independent of the network's traffic:

This case is approximately valid if the acknowledgments are considered as high priority protocols when passing through the network and at each node are inserted into the first packet on the next link on a path to the right destination.[+] When we consider the next case, we will see that the assumption in this case is important in order to maintain a stable situation in the network. With the assumption made here, and for a given routing variable set ϕ, Eq. (4.3) and (4.5) completely describe n_k in terms of the input rates r_m, $m = 1,\ldots,K$, through the functions $t_\ell(f_\ell)$, $\ell = 1,\ldots,L$. Therefore we can view (4.3) and (4.5) as the following mapping from $\mathbb{R}^K$ to $\mathbb{R}^K$:

$$\vec{n} = h(\vec{r}) \tag{4.6}$$

where $\vec{n} \in \mathbb{R}^K$ and $\vec{r} \in \mathbb{R}^K$ are two column vectors with the k'th components n_k and r_k respectively.

[+] When there is no packet going on such a link, a special protocol packet should be formed to send the acknowledgment over that link.

Theorem 4.1 : Let $t_\ell(f_\ell)$, $\ell = 1,\ldots,L$, be an increasing and continuously differentiable function on $[0,C_\ell)$. Assume that $t_\ell(0) > 0$ for $\ell = 1,\ldots,L$, and $\vartheta_k \geq 0$ for $k = 1,\ldots,K$. For a given routing matrix Q, define $\mathcal{D}$ as the set of feasible inputs $\vec{r}$, i.e.

$$\mathcal{D} = \{\vec{r} \mid \vec{r} \geq 0 \quad \text{and} \quad \sum_{k=1}^{K} r_k \cdot q_{k\ell} < \xi_\ell \qquad \ell = 1,\ldots,L\}$$

where ξ_ℓ is the effective capacity of link ℓ as defined in section 2.2. Then there exists a one to one correspondence between the points in $\mathcal{D}$ and $h(\mathcal{D})$, i.e.

$$\begin{cases} h(\vec{r}) = \vec{n}\,, & \vec{r} \in \mathcal{D} \\ h(\vec{r}') = \vec{n}, & \vec{r}' \in \mathcal{D} \end{cases} \Rightarrow \vec{r} = \vec{r}'$$

Furthermore, the inverse function $\vec{r} = h^{-1}(\vec{n})$ is continuously differentiable on $h(\mathcal{D})$ and its derivative with respect to $\vec{r}$ is the inverse of $\dfrac{dh(\vec{r})}{d\vec{r}}$.

This theorem is proved in Appendix C.

CASE 2 - Equal priority in the transmission of acknowledgments and other data:

An alternative to our assumption of giving high priority to the service of acknowledgments is to route them through the network and serve them at the links with the same priorities as other traffic. In this case, if we assume that the average length of packets containing acknowledgments is equal to the average length of other packets, it follows that $\theta_{ij} = \tau_{ji}$, $(i,j) \in C_A$. Notice that if (j,i) does not denote an active commodity, still we can define τ_{ji} as the expected travelling time between j and i.

In this case θ_{ij} is not constant and is a function of the traffic in the network. In order to obtain this function, let the source-destina-

tion pair (i,j) correspond to the commodity k. Let us define the row vector $v_k \in \mathbb{R}^L$ with the ℓ'th entry showing the portion of commodity (j,i) which goes through the link ℓ. θ_k can be expressed then as:

$$\theta_k = v_k \cdot \vec{t} \tag{4.7}$$

It is clear that if (j,i) denotes an active commodity, v_k is equal to that row of Q which corresponds to commodity (j,i). Equations (4.3), (4.5) and (4.7) imply that:

$$n_k = \frac{1}{\Gamma} r_k (v_k + q_k) \cdot \vec{t} \qquad k = 1,\ldots,K \tag{4.8}$$

In appendix C, Eq. (4.7) is used to show that when $\theta_{ij} = \tau_{ji}$, $(i,j) \in C_A$, there is no longer a unique correspondence between $\vec{r}$ and $\vec{n}$ and for a given $\vec{n}$ there may exist multiple inputs $\vec{r} \in \mathcal{D}$ satisfying $\vec{n} = h(\vec{r})$.

Theorem 4.1 suggests that in a nerwork where the acknowledgments are given high priority in service, one way of adjusting the input rates to a set of desired values r^*, is to keep the expected number of outstanding packets of each commodity (i,j) at the value corresponding to r^*, namely to have $\vec{n} = \vec{n}^* = h(\vec{r}^*)$.

The window strategy provides an effective means for controlling the value of $\vec{n}$. Suppose that we choose $w_{ij} = n^*_{ij}$ for all $(i,j) \in C_A$. This implies that the number of outstanding packets of any active commodity (i,j) will always be equal to or smaller than $w_{ij} = n^*_{ij}$. If we further assume that on each active commodity (i,j), there are always some messages waiting to enter the network, then as soon as a new acknowledgment arrives at node i, a new packet enters the network and the number of outstanding packets will always stay at $w_{ij} = n^*_{ij}$. Therefore the expected number of out-

standing packets, n_{ij}, will also become equal to the desired value n^*_{ij}. According to theorem 4.1 it follows that:

$$r_{ij} = r^*_{ij} \qquad (i,j)\in C_A$$

If the acknowledgments in the network do not possess high priority in service, theorem 4.1 does not apply. Indeed in this case, as we pointed out earlier, enforcing the value of $\vec{n}$ to the desired value $\vec{n}^*$ through the implementation of the window strategy, does not necessarily guarantee that $\vec{r} = \vec{r}^*$ and in fact might allow multiple points $\vec{r}$ corresponding to the same value of $\vec{n} = \vec{n}^*$. Thus, the window strategy is not necessarily an effective way of adjusting the input rates in this case.

The impact of the above results goes beyond the scope of the implementation of the window strategy for achieving a desired set of input rates. In fact if there are multiple choices for $\vec{r}$ with a given set of windows, the statistical behavior of the network makes it possible for the input rates to oscillate back and forth between several points. Therefore in order to maintain a stable situation in the network, wherever windowing is implemented, the acknowledgments should have high priority in service so that their expected travelling time becomes independent of the level of traffic in the network.

In the foregoing discussion, in order to conclude that $n_{ij} = w_{ij}$, we assumed that there are always some messages waiting to enter the network for commodity (i,j). This assumption is reasonable if for every commodity there is a sufficiently large buffer located before the flow control device where the arriving messages can be queued until they are accepted by the network. Of course, with this buffer, the actual delay of commodity (i,j) is more than τ_{ij}. We have not, however, introduced this additional delay into our JFCR formulation (Eq. 2.1), since this delay depends on τ_{ij} and

as explained in section 2.5, we would like to keep the assigned input rate r_{ij} independent of λ_{ij} (except for the constraint $r_{ij} \leq \lambda_{ij}$).

Notice that even with this buffer, as the assigned input rate r^*_{ij} approaches the corresponding upper bound λ_{ij}, the likelihood that the buffer is empty at a given time increases, and therefore the expected number of outstanding packets, n_{ij}, will become smaller than $w_{ij} = n^*_{ij}$. Accordingly the actual rate r_{ij} will be somewhat less than r^*_{ij} (Fig. 4.1).

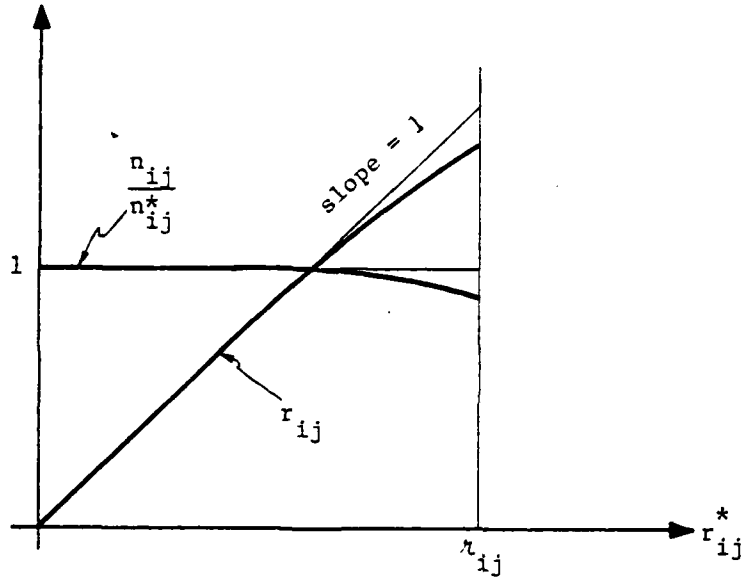


Fig. 4.1

A more important source of error in adjusting r_{ij} to r^*_{ij} for many cases of practical importance is the approximation involved in choosing an integer value for w_{ij} when n^*_{ij} is not integer. This error becomes particularly significant when n^*_{ij} is a fraction of unity. One possible way of correcting this type of error is to change θ_{ij} artificially to some larger value θ'_{ij} for which $n^*_{ij} = \frac{1}{T} r^*_{ij} (\tau_{ij} + \theta'_{ij})$ is an integer. This can be done by disregarding any received acknowledgment at node i (from node j) for $\theta'_{ij} - \theta_{ij}$ seconds.

4.2 Distributed Computation of Window Sizes:

Given a set of assigned input rates r, Eq. (4.5) shows how the appropriate window size $w_{ij}, (i,j)\in C_A$, can be computed. This closed form relationship is inappropriate for a distributed computation of the window sizes. In order to comput w_{ij} distributively, we notice that τ_{ij} can be expressed as follows:

$$\tau_{ij} = \sum_{k:(i,k)\in L} \phi_{ik}(j)\ (t_{ik} + \tau_{kj}) \qquad (4.9)$$

where τ_{kj} denotes the expected travelling time between k and j, for any (k,j) whether it represents an active commodity or not. Naturally, we take $\tau_{jj} = 0$, $j = 1,\ldots,N$.

The similarity between Eq. (4.9) and Eq. (3.7) suggests that τ_{ij} can be computed distributively in the same way that was explained in Section 3.3 for the distributed computation of $\frac{\partial G_T}{\partial r_{ij}}$. This, indeed is possible for a loop-free case, which is the case under our consideration: For any destination j, every node $i \neq j$ should wait until it receives τ_{kj} from all the adjacent and downstream nodes k (downstream with respect to j). Node i itself can measure or calculate the value of t_{ik} for all $(i,k)\in L$. Then it can compute τ_{ij} from (4.9) and in turn send it to all of the adjacent nodes. This process continues until τ_{ij} is known at every node i and for every destination j. Each node i then uses τ_{ij} in adjusting the window size.

In the above explanation we tried to demonstrate that the distributed computation of window sizes is possible. There are, however, some problems involved in this computation that we have neglected. To explain these problems let us consider the network after the m'th iteration of the

algorithm.† Let r^m, ϕ^m, t^m, w_{ij}^m, τ_{kj}^m, $\left(\frac{\partial G_T}{\partial r_{kj}}\right)^m$ for $k,j\in N$, $i:(i,j)\in C_A$ denote the parameters of the network at this stage. For the next iteration it is necessary to comput $w_{ij}^{m+1} = \frac{1}{\Gamma} \cdot r_{ij}^{m+1} \cdot (\tau_{ij}^{m+1} + \theta_{ij})$, for $(i,j)\in C_A$. τ_{ij}^{m+1} is a function of t^{m+1} and ϕ^{m+1} as we have

$$\tau_{ij}^{m+1} = \sum_{k:(i,k)\in L} \phi_{ik}^{m+1}(j)\ (\tau_{kj}^{m+1} + t_{ik}^{m+1}) \qquad (4.10)$$

Therefore, in order to compute τ_{ij}^{m+1}, both ϕ^{m+1} and t^{m+1} are needed. The following procedure describes the distributed computations which are necessary in order to compute τ_{ij}^{m+1} and w_{ij}^{m+1}, $(i,j)\in C_A$.

i) perform the distributed procedure of computing ϕ^{m+1} and r^{m+1}.

ii) every node i should inform all of its adjacent nodes k, whether or not the routing variables $\phi_{ik}^{m+1}(j)$, $j\in N$, $j \neq i$, are zero. Thus every node k will know what its adjacent upstream nodes with respect to different destinations will be in the next iteration.

iii) Perform the following distributed computations to find f_{ik}^{m+1}, $(i,k)\in L$. For each destination j, every node i waits until it receives $f_{ki}^{m+1}(j)$ from all the adjacent upstream nodes k and then obtains $s_{ij}^{m+1} = r_{ij}^{m+1} + \sum_{k:(k,i)\in L} f_{ki}^{m+1}(j)$. Then node i informs all of the adjacent nodes k of the value of

---

† By one iteration we refer to any step of the algorithm in which a new update of the cost differentials $\partial G_T/\partial r_{ij}$, $(i,j)\in C_A$ is necessary. In this sense, one iteration of the algorithm proposed in Section 3.3, corresponds to one application of the mapping A_ϕ or A_r and not to one application of the mapping $A = A_\phi \cdot A_r$.

$f_{ik}^{m+1}(j) = s_{ij}^{m+1} \cdot \phi_{ik}^{m+1}(j)$ for all $j \in N$, $j \neq i$. Once $f_{ik}^{m+1}(j)$ is known for all $(i,k) \in L$, $j \in N$, $i \neq j$, every node i can compute $f_{ik}^{m+1} = \sum_{\substack{j=1 \\ j \neq i}}^{I} f_{ik}^{m+1}(j)$.

iv) Having the value of f_{ik}^{m+1}, each node i can find $t_{ik}^{m+1}(f_{ik}^{m+1})$ either analytically, if the function $t_{ik}(f_{ik})$ is known, or using the following approximation:

$$t_{ik}^{m+1} \approx t_{ik}^{m} + \left(\frac{dt_{ik}}{df_{ik}}\right)^{m} \cdot (f_{ik}^{m+1} - f_{ik}^{m})$$

where t_{ik}^{m} and $\left(\frac{dt_{ik}}{df_{ik}}\right)^{m}$ are estimated by node i.

v) Now the previously explained procedure for computing τ_{ij}^{m+1} based on the values of ϕ^{m+1} and t_{ik}^{m+1} can take place, and then every node i can compute the window sizes $w_{ij}^{m+1} = \frac{1}{\Gamma} r_{ij}^{m+1}(\tau_{ij}^{m+1} + \theta_{ij})$ for all $(i,j) \in C_A$

As is evident, the distributed computations described above require alomost three times the protocol transmission compared to the distributed computations of just updating ϕ and/or r. This increase is due to the additional computations necessary to find τ_{ij}^{m+1}. If one is willing to accept some approximation, it is possible to use a simpler and faster procedure for finding τ_{ij}^{m+1} and w_{ij}^{m+1}, $(i,j) \in C_A$:

As an approximation to τ_{ij}^{m+1}, let us define $\hat{\tau}_{ij}^{m+1}$ as:

$$\hat{\tau}_{ij}^{m+1} = \sum_{k:(i,k)\in L} \phi_{ik}^{m+1}(j) \; (\hat{\tau}_{kj}^{m+1} + t_{ik}^{m}) \quad i, j \in N, \; i \neq j \qquad (4.11)$$

$\hat{\tau}_{ij}^{m+1}$ would be the expected travelling time between i and j, if the routing variable set was ϕ^{m+1} but the link flows were f^m. It is possible to compute $\hat{\tau}_{ij}^{m+1}$ distributively in parallel with the computation of r^{m+1} and ϕ^{m+1}: For every destination j, each node i waits until it receives both $\left(\frac{\partial G_T}{\partial r_{kj}}\right)^m$ and $\hat{\tau}_{kj}^{m+1}$ from all of the adjacent downstream nodes k. Then node i first computes $\phi_{ik}^{m+1}(j)$ for all $(i,k)\in L$, and then obtains the value of $\hat{\tau}_{ij}^{m+1}$ from (4.11). Finally it sends $\hat{\tau}_{ij}^{m+1}$ and $\left(\frac{\partial G_T}{\partial r_{ij}}\right)^m$ to all of the adjacent nodes. The process continues until $\hat{\tau}_{ij}^{m+1}$ and $\phi_{ik}^{m+1}(j)$ are known for all $i,j\in N$, $i \neq j$, $(i,k)\in L$. Then the input rates and window sizes can be updated easily.

Finally an even simpler method of computing the window sizes is to measure the round trip delay $\tau_{ij} + \theta_{ij}$ directly at the node i, and use this value to compute w_{ij} for the next iteration, which means using $\hat{w}_{ij}^{m+1} = \frac{1}{\gamma} \cdot r_{ij}^{m+1} \cdot (\tau_{ij}^m + \theta_{ij})$ as the window size of commodity (i,j) at iteration $(m+1)$. This involves more approximation compared to the previous case since τ_{ij}^m is used to compute w_{ij}^{m+1}. A more serious problem here is that the average round trip delays can be estimated less accurately over a given interval than the link delays since the link flows generally combine many commodities and thus generally contain more packets per unit time.

4.3 Quasi-Static Behavior of the JFCR Strategy

The JFCR strategy and the algorithm developed in chapters II and III are based on the assumption that the input statistics and the link capacities of the network do not vary with time. This assumption is reflected in a fixed set of active commodities C_A and a fixed set of desired input rates λ. The algorithm is intended however for quasi-static applications

where, as time goes on, the desired input rates change slowly and also new commodities become active gradually or some of the active commodities become silent.

While the convergence of the algorithm under these conditions is subject to question, it can be applied to the quasi-static case with slight modification in the procedure of the algorithm. To discuss this necessary modification, consider the network just before an update of the input rates and let C_A and λ respectively represent the set of active commodities and the set of desired input rates at this moment. Let r be assigned as the set of input rates after this update. Until the next update the following quasi-static changes might occur in the network:

a) Some of the desired input rates may increase and new commodities may become active.

b) Some of the desired input rates may decrease. This reduction for some of the commodities may be sufficient to reduce the new desired input rate to less than the assigned rate. At the limit, some of the commodities may become totally silent.

Clearly if case a, b or both occurs, each source node i at the next update of the algorithm should consider the active commodities at that time and should calculate and assign their new input rates, based on the most recent values of the corresponding desired rates (Eq. 3.14). Additional considerations are necessary, however, due to the possibility of case b. Suppose that for some commodity (i,j), the desired input rate in the time interval between two updates decreases from λ_{ij} to some value less than the input rate r_{ij} assigned at the first update. Clearly then the actual input rate of commodity (i,j) in this interval will be less than r_{ij}.

Let r_{ij} (real) denote this actual input rate in contrast with the nominal value r_{ij} assigned by the algorithm. During the interval under consideration, the measurement of the marginal link costs and the cost differentials $\frac{\partial G_T}{\partial r_{ij}}$ is based on the actual traffic passing through the network and not on the traffic specified by the nominal input rates r_{ij}, $(i,j) \in C_A$. Therefore, at the next update of the algorithm, the new input rate r_{ij}^1 should also be computed based on the actual input rate r_{ij}(real) and not the nominal value r_{ij}. Accordingly, Eq. (3.13) and (3.14) should be modified as follows:

$$\delta_{ij} = \left.\frac{\partial J}{\partial r_{ij}}\right|_{r_{ij} = r_{ij}(\text{real})} = \left.\frac{\partial G_T}{\partial r_{ij}}\right/_{r_{ij} = r_{ij}(\text{real})} - \left. p_{ij}(r_{ij})\right/_{r_{ij} = r_{ij}(\text{real})} \tag{4.12}$$

$$r_{ij}^1 = r_{ij}(\text{real}) - \mu\delta_{ij}, \quad 0 \le r_{ij}(\text{real}) - \mu\delta_{ij} \le \pi_{ij}$$

$$= 0 \qquad r_{ij}(\text{real}) - \mu\delta_{ij} \le 0$$

$$= \pi_{ij} \qquad r_{ij}(\text{real}) - \mu\delta_{ij} \ge \pi_{ij} \tag{4.13}$$

Here, π_{ij} denotes the most recent value of the desired input rate.

The variations of r_{ij} and r_{ij}(real) with respect to time are illustrated in Fig. 4.2 for two types of behavior of π_{ij}. In both cases it is assumed that π_{ij} stays below the amount of service which the JFCR strategy can potentially offer. More precisely, it is assumed that δ_{ij} as defined by Eq. (4.12) is always negative. In case a, where π_{ij} changes

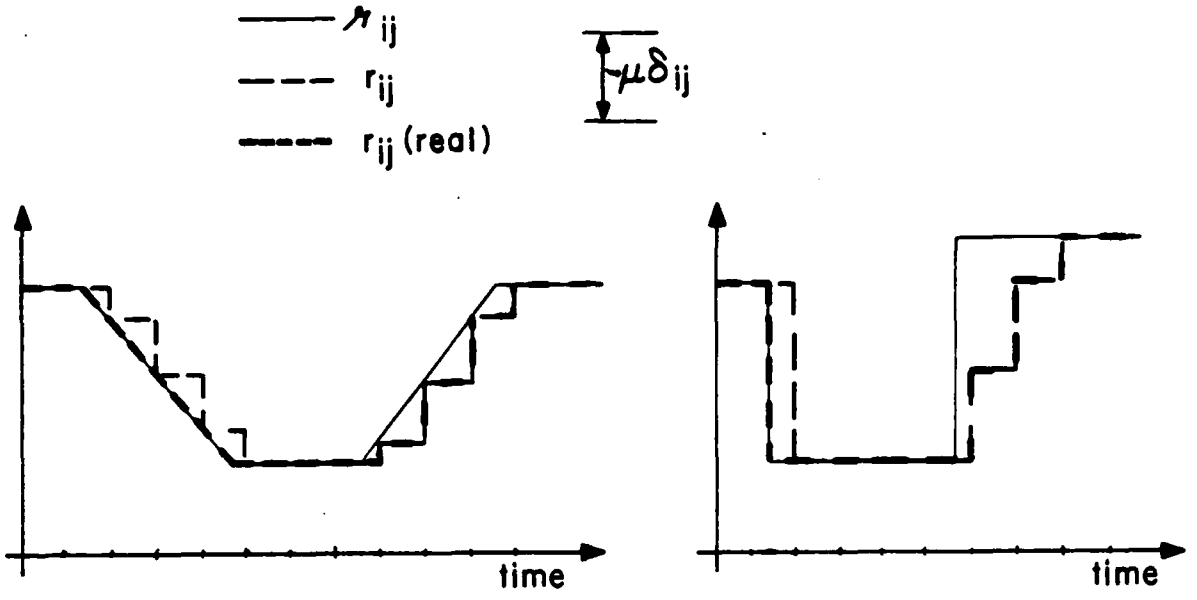


Fig. 4.2

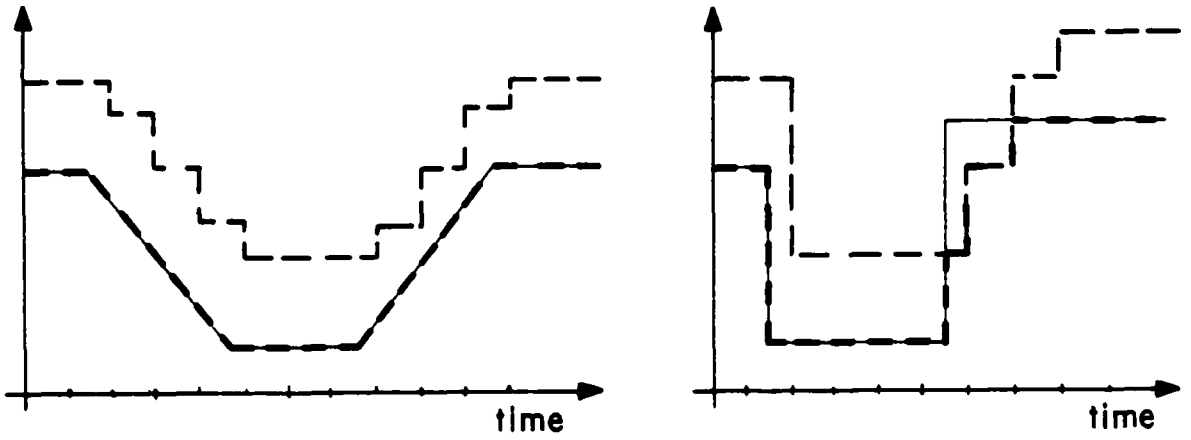


Fig. 4.3

a - π_{ij} changes slowly

b - π_{ij} changes rapidly

gradually with time, the algorithm is able to keep up with it. In case b, when λ_{ij} suddenly increaes, it takes some time until the appropriate input rate is assigned to the commodity, while on the falling edge, r_{ij} follows λ_{ij} almost immediately. This case illustrates an important shortcoming of the proposed JFCR strategy: When a big user of the network, which is inactive for a while, suddenly becomes active, it may take a long time before the proper input rate is assigned to it. To solve this problem, one alternative is to leave the window sizes large when the desired input rates have become too small compared to the active periods. But this has the disadvantage of exposing the network to congestion when many such commodities become active altogether. Further investigations are necessary to find an appropriate way of handling the trade-off between the difficulties of above two alternatives.

The difference between r_{ij} and r_{ij}(real) is not merely caused by the reduction of λ_{ij}. In fact, as we discussed in sections 4.1 and 4.2, there are always some approximations involved in adjusting the input rate by the window strategy. Therefore some difference between r_{ij} and r_{ij}(real) should always be anticipated. In this respect, Fig. 4.2 (and also Fig. 4.3 to be discussed later) are somewhat misleading.

In our discussion in this report, we have assumed so far that the desired input rate λ_{ij} is known by the source node i. While this assumption is reasonable for a static case, for the quasi-static case in which λ_{ij} changes with time, the assumption may not be valid any longer. When node i is updating r_{ij}, if it does not know what λ_{ij} is going to be, it should anticipate some increase in the load offered by the commodity, λ_{ij}, and choose the value of r_{ij} accordingly. Eq. (4.13), therefore, changes to

$$r_{ij}^1 = r_{ij}(\text{real}) - \mu\delta_{ij} \qquad r_{ij}(\text{real}) - \mu\delta_{ij} \geq 0$$

$$= 0 \qquad r_{ij}(\text{real}) - \mu\delta_{ij} \leq 0 \quad (4.14)$$

If r_{ij}^1 as computed by (4.14) is larger than λ_{ij}, the actual input rate $r_{ij}^1(\text{real})$ will be less than r_{ij}^1. Since in the next iteration, r_{ij}^2 is computed based on $r_{ij}^1(\text{real})$, it will be kept closed to λ_{ij}. Fig. 4.3 illustrates how r_{ij} and $r_{ij}(\text{real})$ change according to (4.12) and (4.14), namely, when λ_{ij} is not known by node i.

In summary, we have shown in this section that under quasi-static variations of input statistics and when the offered load λ_{ij} is not known by the source node i, the JFCR strategy can still be implemented and practiced. The essential question of whether it can adapt fast enough to keep up with the changing input statistics is, however, difficult and requires further research. Clearly, in order to keep up with faster statistical variations, the algorithm needs to be updated more frequently. But frequent updating requires more updating protocol which reduces the effective link capacities available for data. It also makes the measurement of marginal link delays and other involved variables less accurate.

4.4 Statistical Fluctuations of the Input Arrivals - Buffer Overflow

An ideal flow control scheme should be able to prevent buffer overflow in the network under all circumstances. That is, it should guarantee that in the course of communication, no packet ever arrives at some node without any buffer space available to store it. The JFCR strategy developed in chapters II and III does not meet such a strong requirement. It only guarantees that if the cost function J is initially limited, then at every stage of the algorithm the expected number of packets waiting at any link

(i,k), namely D_{ik}, remains less than B_{ik}, where B_{ik} is some fraction of the total available buffer space $B_{ik}(max)$. This does not imply that at any instant of time, the number of packets at each link (i,k) is less than B_{ik} or even less than $B_{ik}(max)$.

In order to see this clearly, let us consider an example in which the packets of eqch commodity, at the point where they are admitted by the flow control device into the network, form a poisson process.* In this case it is possible, although very unlikely, that while the expected number of packets allowed to the network is kept at a limited value by the JFCR strategy, a huge number of packets enter the network in a short period of time in which case buffer overflow becomes inevitable at least at some of the nodes. The probability of such an event can be reduced by choosing smaller values for $B_{ik}/B_{ik}(max)$, but will always remain nonzero. We conclude that the JFCR strategy is only capable of reacting to sufficiently slow variations of the input traffic and can not control the more dynamic statistical fluctuations of the input.

This inability to control the short-term statistical variations of the traffic is inherent in the working mechanism of the JFCR strategy. As the congestion builds up in the buffer of some link (i,k),it takes some times until the news value of D_{ik} is measured and the congestion is noticed by the algorithm. It may even take several iterations until sufficient reduction is introduced by the algorithm in the flow of traffic through link (i,k). If the congestion builds up rapidly, however, before these arrange-

---

*As we shall see, this can not happen if the window strategy is employed as the means of adjusting the input rates. However, we consider this case in order to investigate the dynamic behavior of the JFCR strategy alone. The effect of the window strategy will be discussed later.

ments are made, the buffer may overflow.

This limitation of the JFCR strategy indicates that other flow control schemes with faster dynamics should be implemented together with it. The window strategy that was proposed as a mechanism of adjusting the input rates for implementation of the JFCR strategy, in fact is very effective in controlling short-term fluctuations of the arriving traffic as we will see.

The window stategy at any time allows only a limited number of outstanding messages for each commodity. This, in comparison with the example that was just considered, is a big imporvement, since now large bursts of arriving traffic which are going to create congestion will be smoothed out over time by the window strategy before being allowed to enter the network. Moreover, if congestion builds up at some buffer of the network, the input rate of the traffic passing through that buffer will be cut back since these commodities undergo larger amount of delays before reaching their destinations. This can be viewed as a negative feedback effect, in which as congestion builds up, messages arrive at their destinations more slowly, corresponding acknowledgments come back to the source node with a slower rate, and the input rate is cut back accordingly.

It is clear at this point that the proposed window strategy does not simply function as a mechanism of implementing the JFCR strategy but rather plays a distinct and important dynamic flow control role in the network. It is even helpful to view the window strategy as the basis of our proposal for flow control with the JFCR strategy playing the following two roles: First a complementary role in determining the appropriate window sizes based on the quasi-static input statistics, and second, the role of optimal routing of the data in the network.

The argument about the effectiveness of the window strategy in congestion control does not mean, however, that the network is protected against congestion completely. The number of outstanding messages permitted by the window strategy is such that on the average the number of messages waiting on each link (i,k) is less than B_{ik}. But again, the distribution of the outstanding messages on different nodes has a statistical nature and it is possible that many messages cluster at one node creating buffer overlow. The advantage of the window strategy is in making such events very unlikely but the likelihood remains nonzero if the windows are large enough.

The likelihood of buffer overflow in a network where the window strategy is implemented (with the window sizes determined by the JFCR strategy), depends on several factors. First comes the value of $B_{ik}/B_{ik}^{(max)}$ for different links. As this ratio is reduced, the likelihood of buffer overflow at the corresponding link decreases. At the limit, it is always possible to choose a sufficiently small value for $B_{ik}/B_{ik}^{(max)}$ which totally prevents buffer overflow at link (i,k). However, this either would result in very inefficient utilization of the available capacity of link (i,k) (if B_{ik} is small[+]) or would require unreasonably costly buffer assignment (if $B_{ik}^{(max)}$ is very large).

Another important factor in determining the likelihood of buffer overflow in the network is the size of the network and the number of links which a message has to pass through before arriving at the destination. As the communication path's get longer (in terms of the number of links involved), the chance of congestion increases. To see this point clearly consider a

---

[+] Recall from example 2.1 that assuming M/M/1 queues at any link (i,k), the effective capacity of the link would be $\xi_{ik} = C_{ik} \cdot B_{ik}/(1 + B_{ik})$.

commodity which has only one path to the destination with n links. Let w indicate the corresponding window size and for simplicity assume instantaneous acknowledgments. If $n = 1$, namely, if there is only one link which this commodity has to go through, the number of packets waiting at this link is always w and therefore it never exceeds the anticipated average quantity which is w. On the other hand if $n = 20$, assuming that all of the involved links have the same average delay per message, the anticipated average number of messages waiting on each link is $\frac{w}{20}$, while the range of fluctuations of the waiting messages is between 0 and w. Here,obviously we have more chance of congestion unless $B_{ik}(max)$ is much bigger than B_{ik} for all of these links.

Finally, we need to emphasize that in a well-designed network, together with any quasi-static and dynamic flow control scheme (such as the JFCR strategy and the window strategy), the system must have some emergency mode of operation in it, which becomes active whenever the potential for deadlock arises. Some intersting discussions of deadlock recovery systems can be found in [1] and [2].

4.5 Node-to-Node versus End-to-End Flow Control

In this last section, we introduce a rather general type of flow control in comparison with what was discussed previously. In the approach just presented, flow control was viewed merely as an end to end practice. Accordingly, we tried to maintain a noncongesting traffic by imposing restrictions on the traffic, only at the gates of the network. This is not the only type of flow control which can be practiced, nor is it always sufficiently effective for the purpose of avoiding congestion while using the network efficiently, as we showed in the previous section. A more general type of flow control is

to impose restrictions on the traffic on a node-to-node basis. Of course, to be precise we should indicate that minimum cost routing itself is a type of node-to-node flow control which tries to maintain some sort of balance in the level of saturation of different links, so that local congestion does not occur. But in the same way that the input rate assignments of the JFCR strategy were only successful in coping with the long-run variations of the traffic and not with its short-run fluctuations, the routing assignments are only helpful in maintaining balanced traffic on the long-run, and do not effect the problem of local congestion due to the fast fluctuations of traffic.

In order to provide protection against local congestion created by fast fluctuations of traffic, a window strategy on a node to node basis might be used. Presumably the link flow assignments of the JFCR strategy should be used to determine the right size of the node-to-node windows, in which case the node-to-node window sizes could be interpreted as a mechanism of implementing these routing assignments.

The extreme of node-to-node flow control is that over any link (k,ℓ), and for every commodity (i,j) using that link, there be a window size $w_{k\ell}(i,j)$ assigned to node ℓ. This scheme is very costly, however, and it may be desirable to practice node-to-node flow control with less generality. To see the possibility of such a scheme, recall from section 4.4 that as the length of a communication path (in terms of the number of links involved) increases, the chance of local congestion grows. This reveals a more tractable way of viewing and practicing node-to-node flow control : As the size of a network grows, it can be split into smaller subnetworks with the proposed end-to-end flow control strategy practiced for each network.

Further research is necessary to evaluate the need for node-to-node flow control and develop the theory and details of an appropriate strategy.

APPENDIX A

Proof of Lemma 2.2

Assume that at the optimal point (r^*, f^*) there is an active route $R_a(i,j)$ with the length $\lambda_a(i,j) > \lambda_{ij}$. By definition there should be at least one route $R_b(i,j)$ with the length $\lambda_b(i,j) = \lambda_{ij}$. Consider the traffic travelling from node i to node j over route $R_a(i,j)$. Let us change the routing by sending a small part of this traffic, say ε bits per second, over $R_b(i,j)$ and call the new point $(r^*, \tilde{f})$. Therefore,

$$J(r^*,\tilde{f}) - J(r^*,f^*) = G_T(r^*,\tilde{f}) - G_T(r^*,f^*) =$$

$$\sum_{\substack{\text{all links}(\ell,k)\\ \text{in } R_b(i,j)}} [g_{\ell k}(f^*_{\ell k} + \varepsilon) - g_{\ell k}(f^*_{\ell k})] + \sum_{\substack{\text{all links}(\ell,k)\\ \text{in } R_a(i,j)}} [g_{\ell k}(f^*_{\ell k}) - g_{\ell k}(f^*_{\ell k} - \varepsilon)]$$

Since the cost functions $g_{\ell k}(f_{\ell k})$, $(\ell,k)\varepsilon L$, are twice differentiable at $f^*_{\ell k}$, we can use the first order Taylor expansion as follows:

$$J(r^*,\tilde{f}) - J(r^*,f^*) = \sum_{\substack{\text{all links}(\ell,k)\\ \text{in } R_b(i,j)}} \frac{dg_{\ell k}(f^*_{\ell k})}{d f_{\ell k}} \cdot \varepsilon - \sum_{\substack{\text{all links}(\ell,k)\\ \text{in } R_a(i,j)}} \frac{dg_{\ell k}(f^*_{\ell k})}{d f_{\ell k}} \cdot \varepsilon + O(\varepsilon^2)$$

$$= \lambda_b(i,j) \cdot \varepsilon - \lambda_a(i,j) \cdot \varepsilon + O(\varepsilon^2) = (\lambda_{ij} - \lambda_a(i,j)) \cdot \varepsilon + O(\varepsilon^2)$$

Then for a sufficiently small value of ε, $J(r^*,\tilde{f}) - J(r^*,f^*)$ becomes negative which contradicts the assumption that (r^*,f^*) is an optimal point. Therefore at the optimal point (r^*,f^*), the length of any active route $R(i,j)$ is λ_{ij}.

Next consider any commodity $(i,j)\in C_A$ and suppose $r^*_{ij} < \hbar_{ij}$. Let R(i,j) be any route with the length λ_{ij}. Let us define a new point $(\tilde{r},\tilde{f})$ by increasing r^*_{ij} slightly and sending the increased part of r_{ij} over route R(i,j). Using an argument similar to the above, we can show that

$$\lambda_{ij} \geq p_{ij}(r^*_{ij}) \qquad r^*_{ij} < \hbar_{ij} \tag{A.1}$$

Similarly if $r_{ij}{}^* > 0$, take any active route R(i,j) over which part of the nodal flow s_{ij} is passing. Define a new point $(\tilde{r},\tilde{f})$ by decreasing r^*_{ij} slightly and reflecting this decrease only in the flow passing over R(i,j). Then in a similar way it is possible to show that

$$\lambda_{ij} \leq p_{ij}(r^*_{ij}) \qquad r^*_{ij} > 0 \tag{A.2}$$

Combining (A.1) and (A.2), Eq. (2.2) follows. Q.E.D.

Proof of Sufficiency of Theorem 2.1 :

First we establish the following lemma:

Lemma A.1 For any feasible points (r^*,f^*) which satisfies conditions of theorem 2.1 and any other feasible point (r,f), the following inequality holds true with equality if $(r,f) = (r^*,f^*)$

$$\sum_{\substack{(i,k)\in L \\ j\in N}} f_{ik}(j)\, g'_{ik}(f^*_{ik}) \geq \sum_{(i,j)\in C_A} r_{ij}\, \beta_{ij} \tag{A.3}$$

Proof: Multiply both sides of (2.4) by $f_{ik}(j)$ to get

$$(g'_{ik}(f^*_{ik}) + \beta_{kj}) \cdot f_{ik}(j) \geq \beta_{ij}\, f_{ik}(j) \qquad (i,k)\in L,\ j\in N \tag{A.4}$$

(A.4) becomes an equality for $f_{ik}(j) = f^*_{ik}(j)$, since in this case either $f^*_{ik}(j) > 0$ or $f_{ik}(j) = 0$. Summing up (A.4) over i,k and j we get:

$$\sum_{\substack{(i,k)\in L \\ j\in N}} g'_{ik}(f^*_{ik})\cdot f_{ik}(j) \overset{(1)}{\geq} \sum_{\substack{(i,k)\in L \\ j\in N}} \beta_{ij} f_{ik}(j) - \sum_{\substack{(i,k)\in L \\ j\in N}} \beta_{kj} f_{ik}(j)$$

$$= \sum_{i,j} \beta_{ij} \sum_{k:(i,k)\in L} f_{ik}(j) - \sum_{k,j} \beta_{kj} \sum_{i:(i,k)\in L} f_{ik}(j)$$

$$= \sum_{i,j} \beta_{ij} \left[\sum_{k:(i,k)\in L} f_{ik}(j) - \sum_{m:(m,i)\in L} f_{mi}(j) \right] = \sum_{i,j} \beta_{ij} r_{ij} = \sum_{(i,j)\in C_A} \beta_{ij} r_{ij}$$

with equality in (1) for $f = f^*$. Q.E.D.

Now let there be a set of positive numbers β_{ij} satisfying (2.4) and (2.5) at a feasible point (r^*, f^*). For any other feasible point (r, f) and any $0 \leq \alpha \leq 1$, $(\alpha r + (1-\alpha)r^*, \alpha f + (1-\alpha)f^*)$ is also a feasible point. Define $J(\alpha) = J(\alpha r + (1-\alpha)r^*, \alpha f + (1-\alpha)f^*)$. Since $J(\alpha)$ is a convex function,

$$J(r,f) - J(r^*,f^*) = J(1) - J(0) \geq \left.\frac{dJ}{d\alpha}\right|_{\alpha=0} =$$

$$\sum_{(i,j)\in C_A} [(r^*_{ij} - r_{ij}) p_{ij}(r^*_{ij})] + \sum_{\substack{(i,k)\in L \\ j\in N}} [(f_{ik}(j) - f^*_{ik}(j)) g'_{ik}(f^*_{ik})]$$

According to (A.3)

$$\geq \sum_{(i,j)\in C_A} [(r^*_{ij} - r_{ij}) p_{ij}(r^*_{ij})] + \sum_{(i,j)\in C_A} r_{ij}\beta_{ij} - \sum_{(i,j)\in C_A} r^*_{ij}\beta_{ij}$$

$$= \sum_{(i,j)\in C_A} (r^*_{ij} - r_{ij})(p_{ij}(r^*_{ij}) - \beta_{ij}) = \sum_{\substack{(i,j)\in C_A \\ r^*_{ij} = 0}} (r^*_{ij} - r_{ij})(p_{ij}(r^*_{ij}) - \beta_{ij}) +$$

$$\sum_{\substack{(i,j)\in C_A \\ 0<r^*_{ij}<\pi_{ij}}} (r^*_{ij} - r_{ij})(p_{ij}(r^*_{ij}) - \beta_{ij}) + \sum_{\substack{(i,j)\in C_A \\ r_{ij}=\pi_{ij}}} (r^*_{ij} - r_{ij})(p_{ij}(r^*_{ij}) - \beta_{ij})$$

According to (2.5), the second summation above is zero, and the other two summations are both positive since $0 \leq r_{ij} \leq \ell_{ij}, (i,j) \in C_A$. Therefore, $J(r,f) - J(r^*,f^*) \geq 0$ for all feasible (r,f). Q.E.D.

APPENDIX B

We prove theorem 3.5 through establishing the following 3 lemmas:

Lemma B.1 : Let J_o be any positive number. There are scale factors $\mu > 0$ and $\eta > 0$ such that for any feasible point (r^o,ϕ^o) satisfying $J(r^o,\phi^o) \leq J_o$, we have:

$$J(r^1,\phi^1) - J(r^o,\phi^o) \leq 0 \quad \text{for all } (r^1,\phi^1) \in A(r^o,\phi^o)$$

Proof: First consider the point $(r^1,\phi^o) = A_r(r^o,\phi^o)$ and let us define $r^\alpha = \alpha . r^1 + (1 - \alpha) r^o$ and $J(\alpha) = J(r^\alpha,\phi^o)$, $0 \leq \alpha \leq 1$. Since $J(r,\phi)$ is twice differentiable in terms of r, $J(\alpha)$ is also twice differentiable in terms of α. Therefore from the Taylor remainder theorem:

$$J(\hat{\alpha}) - J(0) = \hat{\alpha} \cdot \left.\frac{dJ(\alpha)}{d\alpha}\right|_{\alpha=0} + \frac{1}{2}\hat{\alpha}^2 \left.\frac{d^2J(\alpha)}{d\alpha^2}\right|_{0 \leq \alpha^* \leq \hat{\alpha}} \tag{B.1}$$

$$\frac{dJ(\alpha)}{d\alpha} = \left.\sum_{(i,j) \in C_A} (r_{ij}^1 - r_{ij}^o) \frac{\partial J}{\partial r_{ij}}\right|_{r=r^\alpha} \tag{B.2}$$

$$\frac{d^2J(\alpha)}{d\alpha^2} = \left.\sum_{(i,j) \in C_A} (r_{ij}^1 - r_{ij}^o) \sum_{(\ell,m) \in C_A} (r_{\ell m}^1 - r_{\ell m}^o) \frac{\partial^2 J}{\partial r_{\ell m} \partial r_{ij}}\right|_{r=r^\alpha} \tag{B.3}$$

... (3.13) and (3.14) that:

$$\left.\frac{\partial J}{\partial r_{ij}}\right|_{r_{ij}=r_{ij}^{o}} \cdot (r_{ij}^{1} - r_{ij}^{o}) \leq -\frac{1}{\mu}(r_{ij}^{1} - r_{ij}^{o})^2 \qquad (B.4)$$

It follows from (B.2) and (B.4) that:

$$\left.\frac{dJ(\alpha)}{d\alpha}\right|_{\alpha=0} \leq -\frac{1}{\mu} \sum_{(i,j)\in C_A} (r_{ij}^{1} - r_{ij}^{o})^2 \qquad (B.5)$$

Next let us define the bound M_{J_o} as:

$$M_{J_o} = \max_{(r,\phi):J(r,\phi)\leq J_o} \left|\left|\frac{d^2J(r,\phi)}{dr^2}\right|\right|$$

where $\frac{d^2J(r,\phi)}{dr^2}$ is the Hessian of $J(r,\phi)$ with respect to r, considering r as a vector (the order of components r_{ij} in r is not important) and $\left|\left|\frac{d^2J(r,\phi)}{dr^2}\right|\right|$ is defined as $\max v^t \cdot \frac{d^2J(r,\phi)}{dr^2} \cdot v$ over all vectors v of proper dimension and magnitude one. The bound M_{J_o} exists since $J(r,\phi)$ is twice differentiable in terms of r for $J(r,\phi) < \infty$. It follows from (B.3) that given $J(\alpha) \leq J_o$:

$$\frac{d^2 J(\alpha)}{d\alpha^2} \leq M_{J_o} \cdot \sum_{(i,j)\in C_A} (r_{ij}^{1} - r_{ij}^{o})^2 \qquad (B.6)$$

We know that $J(0) = J(r^o,\phi^o) \leq J_o$. Take any $\hat{\alpha}\in[0,1]$ satisfying:

$$J(\alpha) \leq J(0) \quad \text{for} \quad \alpha\in[0,\hat{\alpha}] \,. \qquad (B.7)$$

According to (B.1), (B.5) and (B.6):

$$J(\hat{\alpha}) - J(0) \leq \sum_{(i,j)\in C_A} (r_{ij}^{1} - r_{ij}^{o})^2 \left[-\frac{1}{\mu}\hat{\alpha} + \frac{1}{2} M_{J_o} \cdot \hat{\alpha}^2\right] \qquad (B.8)$$

For $u < 2/M_{J_o}$ and $\hat{\alpha}\in[0,1]$ the R.H.S. of (B.8) is negative, therefore $J(\hat{\alpha}) \leq J(0)$ with strict inequality if $r^1 \neq r^o$. Since $J(0) \leq J_o$, it follows that (B.7) is true for any $0 \leq \hat{\alpha} \leq 1$. Taking $\hat{\alpha} = 1$ we have the following

with equality if and only if $r^1 = r^o$;

$$J(1) - J(0) = J(r^1,\phi^o) - J(r^o,\phi^o) \leq \left[\sum_{(i,j)\in C_A} (r^1_{ij} - r^o_{ij})^2\right] (\tfrac{1}{2} M_{J_o} - \tfrac{1}{\mu}) \leq 0 \qquad (B.9)$$

Finally, let us consider the effect of the mapping A_ϕ on the cost J. Let $(r^1,\phi^1)\in A_\phi(r^1,\phi^o)$. Since $G_T(r^1,\phi^o) \leq J(r^1,\phi^o) \leq J_o$, it follows from Gallager [4], Appendix C, that there exists an $\eta > 0$ such that

$$J(r^1,\phi^1) - J(r^1,\phi^o) = G_T(r^1,\phi^1) - G_T(r^1,\phi^o) \leq \frac{1}{2\eta(N-1)^3} \sum_{\substack{i,j\in N \\ i\neq j}} \Delta_i^2(j)\, s_{ij}^2 \leq 0$$

where $\Delta_i(j) = \sum_{k:(i,k)\in L} \Delta_{ik}(j)$ (B.10)

and $\Delta_{ik}(j)$ and s_{ij} are the values corresponding to point (r^1,ϕ^o). Summing up (B.9) and (B.10) we get $J(r^1,\phi^1) - J(r^o,\phi^o) \leq 0$.

Q.E.D.

Lemma B.2 Let the scale factors μ and η be chosen as required by lemma B.1 for some given value J_o and let (r^o,ϕ^o) be any feasible point which does not minimize (3.6) and $J(r^o,\phi^o) \leq J_o$. Then

$$J(r^{N-1},\phi^{N-1}) < J(r^o,\phi^o) \quad \text{for all } (r^{N-1},\phi^{N-1})\in A^{N-1}(r^o,\phi^o) \qquad (B.11)$$

Proof: Consider any point $(r^{N-1},\phi^{N-1})\in A^{N-1}(r^o,\phi^o)$. Let (r^n,ϕ^n), n = 1,..,N-2, by any sequence of points satisfying:

$$(r^n,\phi^n)\in A(r^{n-1},\phi^{n-1}) \qquad n = 1,2,\ldots,N-1$$

It follows from Eq. (B.9) and(B.10) that:

$$J(r^n,\phi^n) \leq J(r^n,\phi^{n-1}) \leq J(r^{n-1},\phi^{n-1}) \qquad n = 1,\ldots,N-1$$

Therefore, the only case in which (B.11) does not hold is when we have:

$$J(r^n,\phi^n) = J(r^n,\phi^{n-1}) = J(r^{n-1},\phi^{n-1}) \qquad n=1,\ldots N-1 \qquad (B.12)$$

We show this can happen only if (r^o, ϕ^o) minimizes (3.6), contrary to our assumption:

Let (B.12) hold true. It follows then from (B.9) and (B.10) that $r^{N-1} = r^{N-2} = \ldots = r^o$ and at any point $(r^{n-1},\phi^{n-1}) = (r^o,\phi^{n-1})$, $n=1,\ldots,N-1$,

$$\Delta_{ik}^{n-1}(j) \cdot s_{ij}^{n-1} = 0 \qquad i,k,j\varepsilon N, \quad i \neq j \qquad (B.13)$$

where Δ_{ik}^{n-1} is the value defined by Eq. (17) - (19) corresponding to the mapping of point (r^o,ϕ^{n-1}) to point (r^o,ϕ^n). We show first that if (B.13) holds, there can be no blocking for the mapping of (r^o,ϕ^{n-1}) into (r^o,ϕ^n) for $n = 1,\ldots N-1$. If any blocking occurs, there is some ℓ,m,j for which (3.15) and (3.16) is satisfied (with $\geq$). Thus $s_{\ell j}^{n-1} > 0$ and $\phi_{\ell m}^{n-1}(j) > 0$. Also from (3.18), $a_{\ell m}(j) \geq g'_{\ell m} + \frac{\partial G_T}{\partial r_{mj}} - \frac{\partial G_T}{\partial r_{\ell j}} \geq g'_{\ell m} > 0$. Thus $\Delta_{\ell m}^{n-1}(j) > 0$ and $\Delta_{\ell m}^{n-1}(j) \cdot s_{\ell j}^{n-1} > 0$, which is in contradiction with (B.13).

Next let us denote by $K_{min}^{n-1}(i,j)$, n=1..N-1, the set of points k which achieve the minimization in Eq. (3.18) at the point (r^o,ϕ^{n-1}). It can be seen from Eq. (3.18) - (3.20) that $\phi_{ik}^n(j)$ is nonzero only if $k\varepsilon K_{min}^{n-1}(i,j)$. This is because, according to (B.13), either $\Delta_{ik}^{n-1}(i) = 0$ or $s_{ij}^{n-1} = 0$. If $\Delta_{ik}^{n-1}(j) = 0$, Eq. (3.18) and (3.19) imply that either $k\varepsilon K_{min}(i,j)$, or $\phi_{ik}^{n-1}(j)=0$ in which case $\phi_{ik}^n(j) = \phi_{ik}^{n-1}(j) - \Delta_{ik}^{n-1}(j) = 0$. On the other hand if $s_{ij}^{n-1} = 0$, it follows from Eq. (3.18) and (3.19) that either $k\varepsilon K_{min}^{n-1}(i,j)$ or

$\Delta_{ik}^{n-1}(j) = \phi_{ik}^{n-1}(j)$ in which case $\phi_{ik}^{n}(j) = \phi_{ik}^{n-1}(j) - \Delta_{ik}^{n-1}(j) = 0$. Thus in all cases $\phi_{ik}^{n}(j)$ is nonzero only for $k \epsilon K_{min}^{n-1}(i,j)$.

Now consider a fixed destination j in the network and let us denote by $I_m(j)$ the set of nodes i which are m hops away from j on a shortest route $R(i,j)$ with $g'_{k\ell}$ ($f_{k\ell}$) considered as the length of link (k,ℓ). Notice that according to (B,13) the link flows are the same for all of the points (r^o,ϕ^{n-1}), $n = 1,..N$; therefore the shortest routes and the sets $I_m(j)$ are identical at different steps of algorithm A. From our previous result, for any $i \epsilon I_1(j)$, $\phi_{ik}(j) > 0$ only if $k \epsilon K_{min}^{o}(i,j)$. Since (i,j) in this case is a shortest route from i to j, it also follows that at (r^o, ϕ^1), $\frac{\partial G_T}{\partial r_{ij}} = \lambda_{ij}$. One can also see that $\frac{\partial G_T}{\partial r_{ij}}$ remains the same in the next steps of A namely at (r^o,ϕ^n), $n = 2,..N-1$. Next we know that for any $i \epsilon I_2(j)$, $\phi_{ik}^{2}(j) > 0$ only if $k \epsilon K_{min}^{1}(i,j)$. By definition of $I_2(j)$, at least one element of $K_{min}^{1}(i,j)$ belongs also to $I_1(j)$. Therefore at (r^o,ϕ^2), $\frac{\partial G_T}{\partial r_{ij}} = \lambda_{ij}$. We can continue this argument to show that at (r^o,ϕ^{N-1}), $\frac{\partial G_T}{\partial r_{ij}} = \lambda_{ij}$ for all nodes $i \epsilon N$, since no node can be more than N-1 hops away from j. Thus it follows from Theorem 3.4 that (r^o,ϕ^{N-1}) minimizes (3.6). Since by assumption $J(r^o,\phi^o) = J(r^o,\phi^{N-1})$, (r^o,ϕ^o) also minimizes (3.6). Thus we have shown that (B.11) fails to hold only if (B.12) holds and (B.12) holds only when (r^o, ϕ^o) minimizes (3.6). Q.E.D.

Lemma B.3 The mapping A is a closed mapping (in the sense defined in page 124 of [15]).

Proof: First notice from (3.13) and (3.14) that A_r is a continuous point to point mapping. Thus according to corollary 2, p. 125 of [15], in order

to verify lemma B.3 we only need to show that A is a closed point to set mapping.

Let (r,ϕ^n) be a sequence of feasible points of (3.6) converging to (r,ϕ). Take any $(i,k)\varepsilon L$ and $j\varepsilon N$, $i \neq j$, and let $a^n_{ik}(j)$ be the value of $a_{ik}(j)$ at (r,ϕ^n) as defined by (3.18) for any possible choice of $\hat{\mathcal{B}}^n_{ij}$ and let $a^n_{ik}(j) \to a_{ik}(j)$. Let us denote by $\gamma_{im}(j)$ and $\gamma^n_{im}(j)$, the value of $g'_{im}(f_{im}) + \frac{\partial G_T}{\partial r_{mj}}$ respectively at (r,ϕ) and (r,ϕ^n). According to theorem 3.2, $\gamma_{im}(j)$ is continuous in (r,ϕ); therefore $\gamma^n_{im}(j) \to \gamma_{im}(j)$ for all $(i,m)\varepsilon L$. Since $a^n_{ik}(j)$ approaches a limit, it follows from (3.18) that $\min_{m\notin\hat{\mathcal{B}}^n_{ij}} \gamma^n_{im}(j)$ also approaches a limit, which for a subsequence of n must be achieved at some particular m. Since this m is not blocked for the subsequence, it can be considered not blocked at (r,ϕ). Similarly, if there is any node ℓ which is blocked and $\gamma^n_{i\ell}(j) < \min_{m\notin\hat{\mathcal{B}}^n_{ij}} \gamma^n_{im}(j)$ for this subsequence of n, ℓ can also be blocked for (r,ϕ). Therefore, the mapping from (r,ϕ) to $a_{ik}(j)$ is a closed mapping. Furthermore, since s_{ij} is continuous in (r,ϕ), $\eta \cdot a_{ik}(j)/s_{ij}$ is also a closed point to set mapping from (r,ϕ) for $s_{ij} > 0$. Finally since $0 \leq \phi_{ik}(j) \leq 1$, it follows from (3.19) that $\Delta_{ik}(j)$ is a closed point to set mapping for arbitrary s_{ij}. Thus from (3.17) and (3.20), A_ϕ is a closed point to set mapping.

Q.E.D.

Proof of Theorem 3.5 Let us defins S as the set of points (r,ϕ) which are feasible for (3.6) and $J(r,\phi) \leq J_o$, i.e.

$$S = \{(r,\phi) \mid 0 \leq r_{ij} \leq \kappa_{ij},\ (i,j)\varepsilon C_A, \phi \text{ is a routing variable set}, J(r,\phi) \leq J_o\}$$

Since S is a compact set and A is a closed mapping from S into itself, it follows from corollary 1, p. 124 of [15] that A^{N-1} is also a closed mapping. Theorem 3.5 now follows from lemma B.2 and the general convergence theorem, p. 125 of [15], using A^{N-1} as the algorithm.

Proof of Theorems 3.7 and 3.8

First notice from Eq. (3.21.c) that $\bar{s}_{ij}$ is equal to the corresponding nodal flow in network $\bar{M}$. The routing variables of network $\bar{M}$ are as follows:

$$\bar{\phi}_{ik}(j) = \frac{\bar{f}_{ik}(j)}{\bar{s}_{ij}} = \frac{f_{ik}(j)}{\bar{s}_{ij}} = \psi_{ik}(j) \qquad (i,k)\in L$$

$$\bar{\phi}_{ik'}(j) = \frac{\bar{f}_{ik'}(j)}{\bar{s}_{ij}} = \frac{r_{ij} - r_{ij}}{\bar{s}_{ij}} = \psi_{ij'}(j) \qquad k=j, \quad (i,k)\in C_A$$

$$= 0 \qquad k\neq j, \quad (i,k)\in C_A$$

Therefore, each set ψ uniquely specifies the routing variables and the multicommodity flows of network $\bar{M}$, which in turn correspond to a unique point (r,f) (Th. 3.2). Thus theorem 3.7 is verified.

To prove theorem 3.8, consider the routing algorithm of [17] for network $\bar{M}$. We shall introduce a slight modification in this algorithm to come up with algorithm A_ψ : In order to satisfy constraint (3.3.d), for every $(i,j)\in C_A$, we shall define the set of blocked nodes $\bar{B}_i(j)$ of network $\bar{M}$ to also include all links (i,k'), $k \neq j$. This extension in the definition of $\bar{B}_i(j)$ does not effect the proof of convergence in [17]. Furthermore, according to corollary 3.1, any limit point of the algorithm with the above extension in $\bar{B}_i(j)$, is still an optimal point for the routing problem 3.1.

Next notice that for every node i and destination j, with the extended set $\bar{B}_i(j)$, the following routing variables of network $\bar{M}$, are always zero during the algorithm:

$$\bar{\phi}_{ik}(j) = 0 \qquad k \epsilon N, \quad (i,k) \notin L$$

$$\bar{\phi}_{ik}(j) = 0 \qquad k \neq j, \quad (i,k) \epsilon C_A$$

The subproblem (2.9) of [17] can accordingly be simplified by dropping the terms corresponding to the above routing variables. What remains is the same as the algorithm A_ψ discussed in section 3.4. Therefore, according to theorem 3.1 and the convergence theorem of [17], there exists an $\alpha > 0$ for which algorithm A_ψ converges to a solution of (2.1). Furthermore, according to [16], the JFCR variable set $\psi^m = A_\psi^m(\psi^o)$ corresponds to a loop-free routing for all $m = 1,2,3,\ldots$ Q.E.D.

APPENDIX C

Proof of Theorem 4.1

First we show that the matrix $P = \frac{dh}{d\vec{r}_K}$ (with entries $P_{km} = \frac{\partial n_k}{\partial r_m}$) is nonsingular for $\vec{r} \epsilon D$. We know that $f_\ell = \sum_{k=1} q_{k\ell} \cdot r_k$, for $\ell = 1,\ldots L$. It follows then from (4.3) that:

$$\frac{\partial \tau_k}{\partial r_m} = \sum_{\ell=1}^{L} q_{k\ell} \cdot \frac{\partial t_\ell}{\partial r_m} = \sum_{\ell=1}^{L} q_{k\ell} \cdot \frac{dt_\ell}{df_\ell} \cdot q_{m\ell}$$

$$= q_k \cdot M \cdot q_m^t \qquad k,m = 1,\ldots K \qquad (C.1)$$

where q_m^t denotes the transpose of q_m and M is an LxL diagonal matrix with the entries $M_{\ell\ell} = \frac{dt_\ell}{df_\ell}$ on the diagonal. Also it follows from

(4.5) that:

$$P_{km} = \frac{\partial n_k}{\partial r_m} = \frac{1}{\Gamma}\cdot r_k \cdot \frac{\partial \tau_k}{\partial r_m} \qquad k \neq m$$
$$= \frac{1}{\Gamma}\left[r_k \cdot \frac{\partial \tau_k}{\partial r_m} + (\tau_k+\theta_k)\right] \qquad k=m \tag{C.2}$$

Therefore if P_k denotes the k'th row of P, we have from (C.1) and (C.2) that

$$P_k = \frac{1}{\Gamma}(\tau_k+\theta_k)\cdot e_k + \frac{1}{\Gamma} r_k \cdot q_k \cdot M \cdot \left[q_1^t \mid q_2^t \mid \ldots \mid q_K^t\right] \tag{C.3}$$

where $e_k \in \mathbb{R}^K$ is a row vector with one in the k'th place and zero in other places. Notice that the last matrix in (C.3) is Q^t. Finally it follows from (C.3) that:

$$P = T + R\cdot Q\cdot M\cdot Q^t \tag{C.4}$$

where T and R are KxK diagonal matrices, respectively with the entries $T_{kk} = \frac{\tau_k+\theta_k}{\Gamma}$ and $R_{kk} = \frac{r_k}{\Gamma}$ on the diagonal.

In order to show that P is a nonsingular matrix, we consider the following two cases:

Case a — $r_k > 0$, k=1,2,...K:

In this case P can be expressed in the following form:

$$P = R(T^r + Q\cdot M\cdot Q^t) \tag{C.5}$$

where T^r is a diagonal KxK matrix with the entries $T^r_{kk} = \frac{\tau_k+\theta_k}{r_k}$ on the diagonal. Since $T^r_{kk} > 0$ for k=1,...K, T^r is a positive definite matrix. Since M is a diagonal matrix with nonnegative entries, for any $x \in \mathbb{R}^K$ we have:

$$x^t\cdot(Q\cdot M\cdot Q^t)\cdot x = (Q^t\cdot x)^t \cdot M \cdot (Q^t\cdot x) \geq 0$$

Therefore $Q \cdot M \cdot Q^t$ is positive semidefinite. It follows that $T^r + Q \cdot M \cdot Q^t$ is positive definite, and thus nonsingular. Since R is also nonsingular in this case, it follows from (C.5) that P is nonsingular and det. $P > 0$.

Case b — $r_k = 0$ for some commodities:

For any k with $r_k = 0$, the corresponding row in the matrix $R \cdot Q \cdot M \cdot Q^t$ is zero. It follows then from Eq. (C.4) that for any k with $r_k = 0$, the corresponding row of P only has the diagonal entry $P_{kk} = \frac{\theta_k + \tau_k}{\Gamma}$. Let us construct a matrix $\hat{P}$ by eliminating the k'th row and the k'th column of P for all k with $r_k = 0$. It can be seen from (C.4) that:

$$\hat{P} = \hat{T} + \hat{R} \cdot \hat{Q} \cdot M \cdot \hat{Q}^t$$

where $\hat{T}$ and $\hat{R}$ are constructed by eliminating the corresponding rows and columns of T and R, and $\hat{Q}$ is constructed by eliminating the corresponding rows of Q. Since $\hat{R}$ has only nonzero elements on the diagonal, according to Case 1, $\det \hat{p} > 0$. Therefore,

$$\det P = \det \hat{P} \cdot \prod_{\substack{\text{all } k \\ \text{with } r_k = 0}} \left(\frac{\theta_k + \tau_k}{\Gamma}\right) > 0$$

Therefore the matrix $P = \frac{dh}{d\vec{r}}$ is nonsingular for any $\vec{r} \in D$. It follows from the implicit function theorem [19] that for any point $\vec{r}_o \in D$ and $\vec{n}_o = h(\vec{r}_o)$, there exist small spheres $||\vec{r} - \vec{r}_o|| < \varepsilon$, $||\vec{n} - \vec{n}_o|| < \delta$ in which there is a one to one correspondence between $\vec{r}$ and $\vec{n}$. Furthermore, the resulting implicit function $\vec{r} = h^{-1}(\vec{n})$ is continuously differentiable on $h(D)$ and its derivative with respect to $\vec{n}$ is P^{-1}.

Notice that the one to one correspondence between $\vec{r}$ and $\vec{n}$, as claimed in theorem 4.1, is a stronger condition than we have established up to here. In order to complete the proof of theorem 4.1, we need to

show that the one to one correspondence between $\vec{r}$ and $\vec{n}$ is valid in the whole region D and $h(D)$. To do so, let us consider two arbitrary points $\vec{r}$ and $\vec{r}'$ in D. Let $\vec{n} = h(\vec{r})$ and $\vec{n}' = h(\vec{r}')$. Similarly in the following, any parameter which is primed, corresponds to the imput $\vec{r}'$. We have from Eq. (4.5) that:

$$n_k' - n_k = \frac{1}{\Gamma}[r_k'(\tau_k'+\theta_k) - r_k(\tau_k+\theta_k)] = \frac{1}{\Gamma}[(r_k'-r_k) \cdot (\tau_k'+\theta_k) + r_k(\tau_k'-\tau_k)] \tag{C.6}$$

From Eq. (4.3) we have:

$$\tau_k' - \tau_k = \sum_{\ell=1}^{L} q_{k\ell}\cdot[t_\ell(f_\ell') - t_\ell(f_\ell)] \tag{C.7}$$

According to the mean value theorem, for each link ℓ, there exists a point f_ℓ^* on the line segment connecting f_ℓ and f_ℓ' such that:

$$t_\ell(f_\ell') - t_\ell(f_\ell) = \frac{dt_\ell}{df_\ell}\Big/_{f_\ell = f_\ell^*} \cdot (f_\ell'-f_\ell) \tag{C.8}$$

It follows from (C.7) and (C.8) that

$$\tau_k' - \tau_k = \sum_{\ell=1}^{L} q_{k\ell} \cdot \frac{dt_\ell}{df_\ell}\Big/_{f_\ell=f_\ell^*} \cdot (f_\ell'-f_\ell) = \sum_{\ell=1}^{L} q_{k\ell} \cdot \frac{dt_\ell}{df_\ell}\Big/_{f_\ell=f_\ell^*} \cdot q_\ell^t(\vec{r}'-\vec{r}) \tag{C.9}$$

If $\vec{\tau} \in \mathbb{R}^K$ and $\vec{\tau}' \in \mathbb{R}^K$ are column vectors made of components τ_k and τ_k', we see from (C.9) that

$$\vec{\tau}' - \vec{\tau} = Q \cdot M^* \cdot Q^t \cdot (\vec{r}'-\vec{r}) \tag{C.10}$$

where M^* is an LxL diagonal matrix with diagonal entries $M_{\ell\ell}^* = \frac{dt_\ell}{df_\ell}\Big/_{f_\ell=f_\ell^*}$.

Finally from (C.6) and (C.10) we conclude that:

$$\vec{n}' - \vec{n} = (T'+R \cdot Q \cdot M^* \cdot Q^t) \cdot (\vec{r}'-\vec{r}) \qquad (C.11)$$

Now consider the matrix $P^* = T'+R \cdot Q \cdot M^* \cdot Q^t$. P^* is the same as P except that the parameters τ_k, k=1,...K, and $\frac{dt_\ell}{df_\ell}$, ℓ=1...L, in matrix P^* are evaluated at different points on the line segment connecting $\vec{r}$ and $\vec{r}'$, rather than being calculated at the same point. However, P^* is also a nonsingular matrix. This is because, in the process of proving the nonsingularity of P, only the structure of P and the fact that the parameters τ_k, k=1,...K, are strictly positive and the parameters $\frac{dt_\ell}{df_\ell}$, ℓ=1...L, are nonnegative were used. Since P^* shares all of these properties with P, the same argument can be restated for P^* to prove that it is nonsingular. Therefore from (C.11):

$$\vec{n}' - \vec{n} = 0 \Rightarrow \vec{r}'-\vec{r} = 0 \qquad \text{Q.E.D.}$$

Case 2 – $\theta_{ij}=\tau_{ji}$, $(i,j)\in C_A$:

We use an example in this case to show that when $\theta_{ij}=\tau_{ji}$, $(i,j)\in C_A$, there can be multiple sets of input rates r corresponding to the same vector $\vec{n}$. Consider the network of Fig. C.1 with two active commodities r_1 (from node 2 to node 5) and r_2 (from node 4 to node 1). Let half of r_1 take the links (2,4) and (4,5) and the other half take (2,3), (3,4) and (4,5); but assume that acknowledgments all take the route (5,4), (4,3) and (3,2). Similarly let half of r_2 take (4,2) and (2,1) and the other half take (4,3), (3,2) and (2,1); but assume that the acknowledgments all take the route (1,2), (2,3) and (3,4). Let the capacities of links (1,2), (2,1), (4,5) and (5,4) be large enough so that the delay of these

links is negligible. Let the other links all have capacity C and the delay function $t(f) = \frac{1}{C-f}$. With these assumptions we have:

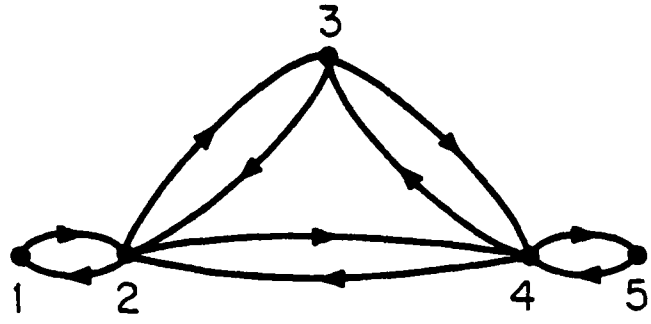


Fig. C.1

$$\tau_1 = 0.5 \cdot \frac{1}{C-r_1} + 0.5 \cdot (+\frac{1}{C-r_1}+\frac{1}{C-r_1}) = \frac{1.5}{C-r_1}$$

$$\theta_1 = \frac{1}{C-r_2} + \frac{1}{C-r_2} = \frac{2}{C-r_2}$$

Similarly: $\tau_2 = \frac{1.5}{C-r_2}$ and $\theta_2 = \frac{2}{C-r_1}$

Therefore: $n_1 = r_1(\frac{1.5}{C-r_1} + \frac{2}{C-r_2})$

and $n_2 = r_2(\frac{1.5}{C-r_2} + \frac{2}{C-r_1})$

If we further assume that $r_1 + r_2 = \frac{7}{4} C$, it follows that:

$$n_1 = n_2 = \frac{r_1 r_2}{2(C-r_1)(C-r_2)} \qquad \text{(c.12)}$$

If we interchange the values of r_1 and r_2, $\vec{n} = (n_1, n_2)$ will remain the same. Therefore, in this case $\vec{r} = (r_1, r_2)$ and $\vec{r}' = (r_2, r_1)$ both correspond to the same $\vec{n}$, if $r_1 + r_2 = \frac{7}{4} C$.

It is not difficult to find the matrix $P = \frac{dh}{d\vec{r}}$ in this case. Similar to Eq. (C.1), one can easily conclude from (4.7) that:

$$\frac{\partial \theta_k}{\partial r_m} = v_k \cdot M \cdot q_m^t \qquad k,m=1,\ldots K \tag{C.13}$$

From (C.1) and (C.13) we have

$$\frac{\partial (\theta_k+\tau_k)}{\partial r_m} = (v_k+q_k) \cdot M \cdot q_m^t \tag{C.14}$$

Using Eq. (4.5) and (C.14), one can proceed in a manner similar to the previous case to find that:

$$P = T+R(Q+V) \cdot M \cdot Q^t \tag{C.15}$$

where V is a KxL matrix composed of rows v_k.

For the network of Fig. C.1 it is possible to deduce from (C.12) that p is singular at $r_1 = r_2 = \frac{7}{8} C$, without computing P from (C.15).

References

1) Raubold E. and J.Haenle, "A Method of Deadlock-Free Resource Allocation and Flow Control in Packet Networks", paper in the Flow Control Session of the International Conference on Computer Communications, Toronto, Canada, Aug. 3-6. 1976.

2) Melin, P.M., P.J. Schweitzer, "Deadlock Avoidance in Store and Forward Networks". Unpublished.

3) Cantor, D.G. and M. Gerla, "Optimal Routing in a Packet Switched Computer Network", IEEE Trans. Computers, Oct. 1974, pp.1062-1069.

4) Gallager, R.G., "A Minimum Delay Routing Algorithm Using Distributed Computation", IEEE Trans. Communications, Jan. 1977, pp. 73-85.

5) Agnew C., "On the Optimality of Adaptive Routing Algorithms", National Telecommunications Conference, 1974, pp. 1021-1025.

6) Yee, J.R., "Minmax Routing in Computer Communication Networks" Sc.D. Thesis, in preparation, M.I.T. Dept. of Electrical Engineering and Computer Science.

7) Vastola, K., "Comparison of Minimum Delay and Min-max Routing Algorithms", M.S. Thesis, Dept. of Electrical Engineering, Univ. of Illinois, June 1979.

8) Kahn, R. and Crowther, W., "Flow Control in a Resource-Sharing Computer Network", IEEE Trans. Communications, June 1972, pp. 539-546.

9) Davies, D.W., "The Control of Congestion in Packet-Switching Networks", IEEE Trans. Communications, June 1972, pp. 546-550.

10) Cerf, V.G. and Kahn, R.E., "A Protocol for Packet Network Intercommunication", IEEE Trans. Communications, May 1974, pp. 637-648.

11) Pennotti, M.C. and M. Schwartz, "Congestion Control in Store and Forward Tandem Links", IEEE Trans. Communications, Dec. 1975, pp. 1434-1443.

12) Lam, S.S. and M. Reiser, "Congestion Control of Store and Forward Networks by Input Buffer Limits", IEEE Trans.Communications, Jan.1979, pp. 127-134.

13) Chatterjee, A., N.D. Georganas, P.K. Verma, "Analysis of a Packet-Switched Network with End-to-End Congestion Control and Random Routing", paper in the Flow Control Sesssion of the International Conference on Computer Communications, Toronto, Canada, Aug. 3-6, 1976.

14) Gerla, M. and W. Chou, "Flow Control Strategies in Packet Switched Computer Networks", NTC Conference Proceedings, San Diego, Cal., Dec. 20,1974.

15) Luenberger, D.G., Introduction to Linear and Nonlinear Programming, Reading, Massachusetts: Addison-Wesley Publishing Company, Inc. 1973.

16) Bertsekas, D.P., "Algorithms for Optimal Routing of Flow in Networks", Coordinated Science Lab. Working Paper, Univ. of Illinois at Urbana-Champaign, June 1978.

17) Gafni, E., "Convergence of a Routing Algorithm", LIDS Report No. 907, M.I.T., May 1979.

18) Baskett, F., K.M.Chandy, R.R. Muntz, F.G. Palacios, "Open, Closed, and Mixed Networks of Queues with Different Classes of Customers" Journal of the Association of Computing Machinery, Vol. 22, No. 2, April 1975, pp. 248-260.

19) Buck, R.C., Advanced Calculus, Section 5.7, New York: McGraw-Hill,Inc. 1976.

Distribution List

Defense Documentation Center Cameron Station Alexandria, Virginia 22314	12 Copies
Assistant Chief for Technology Office of Naval Research, Code 200 Arlington, Virginia 22217	1 Copy
Office of Naval Research Information Systems Program Code 437 Arlington, Virginia 22217	2 Copies
Office of Naval Research Branch Office, Boston 495 Summer Street Boston, Massachusetts 02210	1 Copy
Office of Naval Research Branch Office, Chicago 536 South Clark Street Chicago, Illinois 60605	1 Copy
Office of Naval Research Branch Office, Pasadena 1030 East Greet Street Pasadena, California 91106	1 Copy
New York Area Office (ONR) 715 Broadway - 5th Floor New York, New York 10003	1 Copy
Naval Research Laboratory Technical Information Division, Code 2627 Washington, D.C. 20375	6 Copies
Dr. A. L. Slafkosky Scientific Advisor Commandant of the Marine Corps (Code RD-1) Washington, D.C. 20380	1 Copy

Office of Naval Research
Code 455
Arlington, Virginia 22217
1 Copy

Office of Naval Research
Code 458
Arlington, Virginia 22217
1 Copy

Naval Electronics Laboratory Center
Advanced Software Technology Division
Code 5200
San Diego, California 92152
1 Copy

Mr. E. H. Gleissner
Naval Ship Research & Development Center
Computation and Mathematics Department
Bethesda, Maryland 20084
1 Copy

Captain Grace M. Hopper
NAICOM/MIS Planning Branch (OP-916D)
Office of Chief of Naval Operations
Washington, D.C. 20350
1 Copy

Mr. Kin B. Thompson
Technical Director
Information Systems Division (OP-91T)
Office of Chief of Naval Operations
Washington, D.C. 20350
1 Copy

Advanced Research Projects Agency
Information Processing Techniques
1400 Wilson Boulevard
Arlington, Virginia 22209
1 Copy

Dr. Stuart L. Brodsky
Office of Naval Research
Code 432
Arlington, Virginia 22217
1 Copy

Captain Richard L. Martin, USN
Commanding Officer
USS Francis Marion (LPA-249)
FPO New York 09501
1 Copy